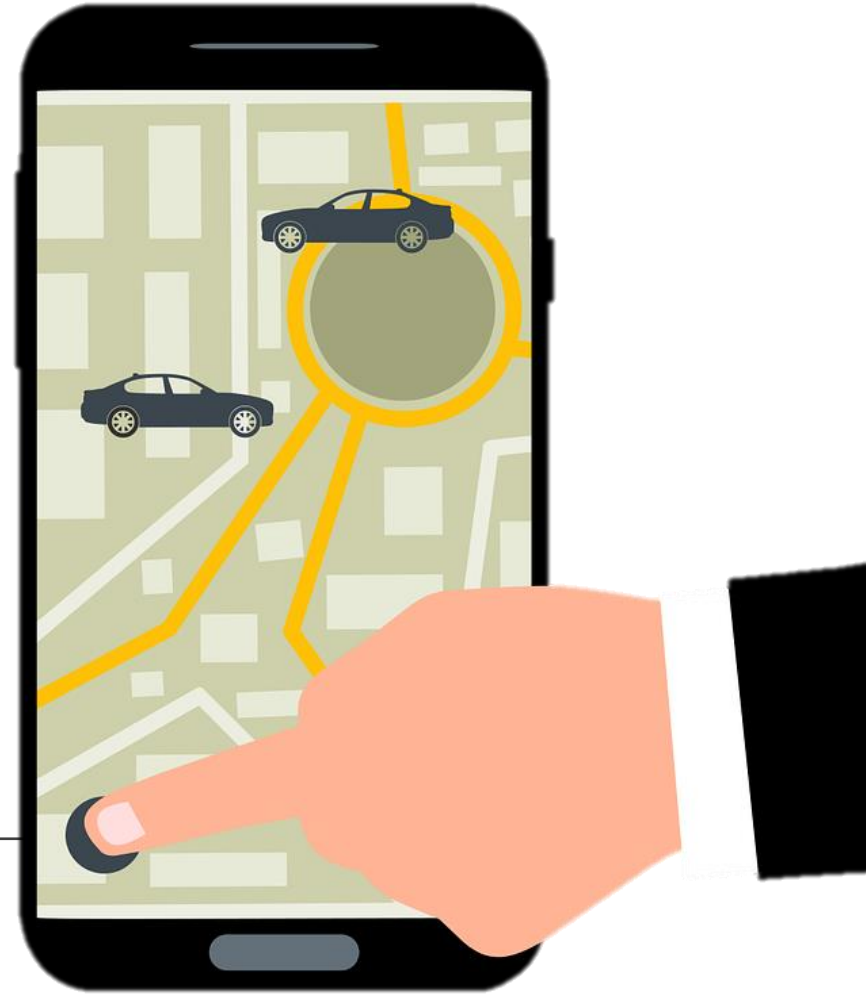
Universidad Autónoma de Madrid

Escuela Politécnica Superior



Implementación de un sistema de estimaciones

del tiempo de recogida para taxis en un entorno de baja latencia y alta disponibilidad

---

TRABAJO FIN DE MÁSTER

Adrián Otero Rodríguez

adrian.otero@estudiante.uam.es

CONTENIDO

1 INTRODUCCIÓN

2 PROBLEMA PLANTEADO

3 DISEÑO Y DESARROLLO

4 INTEGRACIÓN Y RESULTADOS

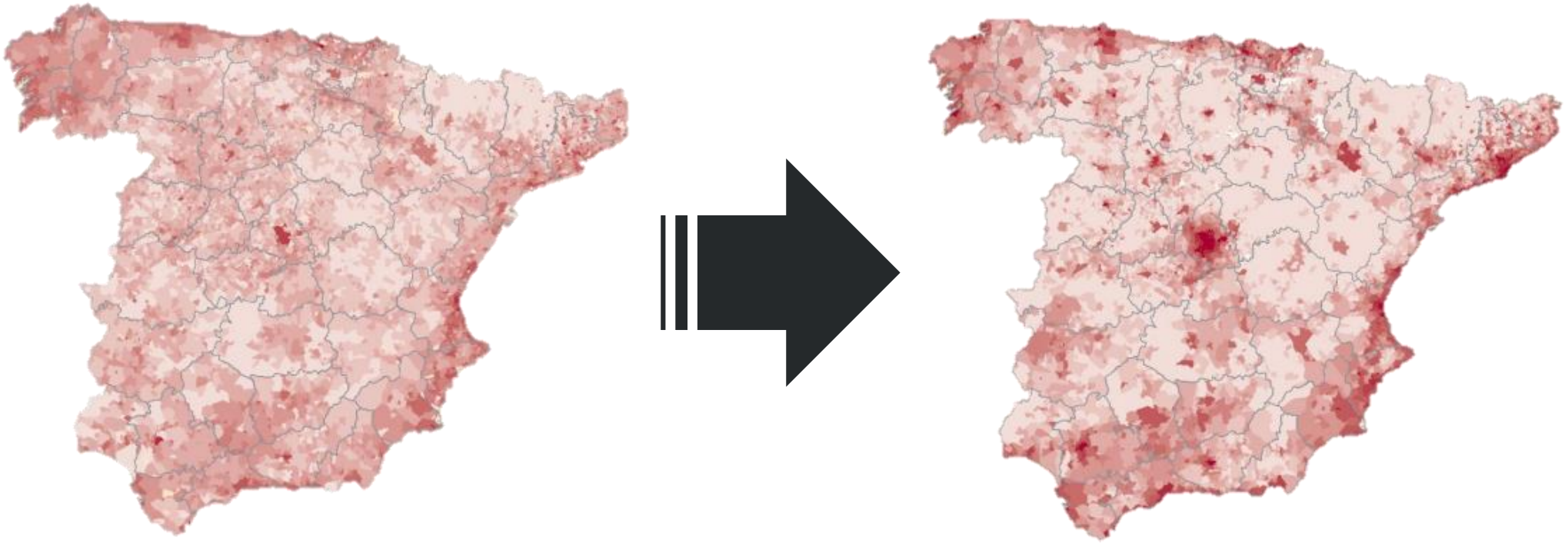
5 CONCLUSIONES Y TRABAJO FUTURO

1

INTRODUCCIÓN

1 INTRODUCCIÓN

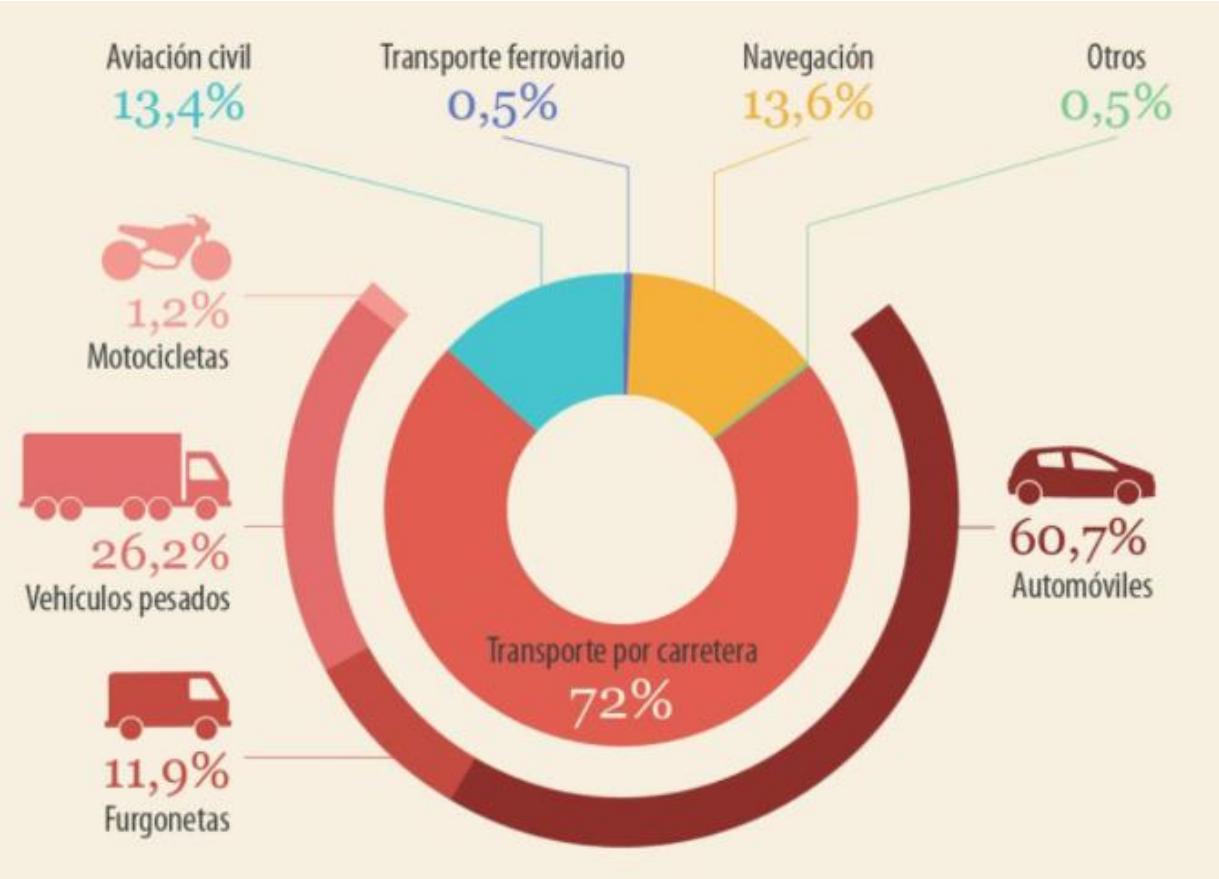
"La despoblación es un fenómeno demográfico y territorial, que consiste en la disminución del número de habitantes de un territorio o núcleo con relación a un período previo"



Año 1900

Año 2001

1 INTRODUCCIÓN



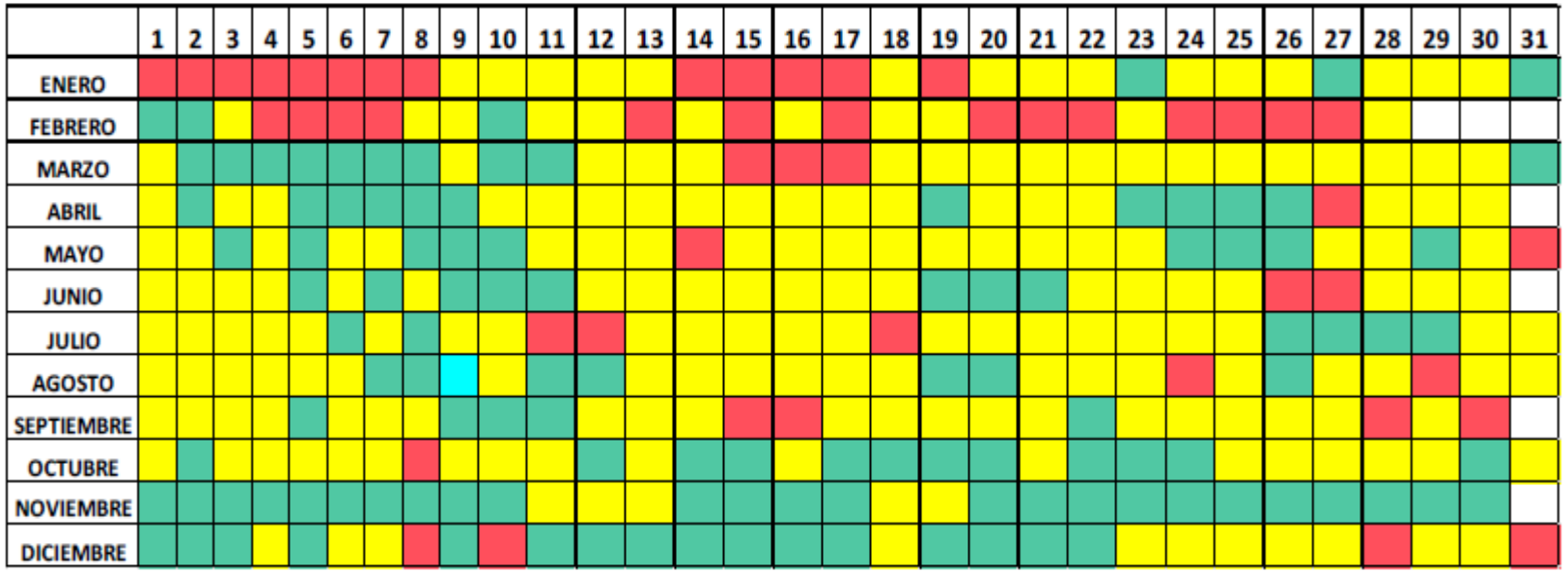
"Las emisiones de CO2 en el transporte de pasajero varían significativamente según el medio"

"Los coches son el principal contaminante, con un 60,7% del total de las emisiones del transporte en carretera de Europa"

Fuente: Parlamento Europeo. "Emisiones de CO2 de los coches: hechos y cifras"

1 INTRODUCCIÓN

"El dióxido de nitrógeno (NO2) es un contaminante indicador de actividades de transporte, especialmente el tráfico rodado. Lo emiten directamente los vehículos, especialmente los diésel (emisiones directas o "primarias")"



Índice Nacional de Calidad del Aire (Orden TEC/351/2019, de 18 de marzo)

CALIDAD DEL AIRE	Índice de Calidad del Aire($\mu g/m^3$)				
Contaminantes	**Muy bueno**	**Bueno**	**Regular**	**Malo**	**Muy malo**
Partículas PM2.5	0-10	11-20	21-25	26-50	51-800
Partículas PM10	0-20	21-35	36-50	51-100	101-1200
Dióxido de Nitrógeno (NO_2)	0-40	41-100	101-200	201-400	401-1000
Ozono (O_3)	0-80	81-120	121-180	181-240	241-600
Dióxido de Azufre (SO_2)	0-100	101-200	201-350	351-500	501-1250

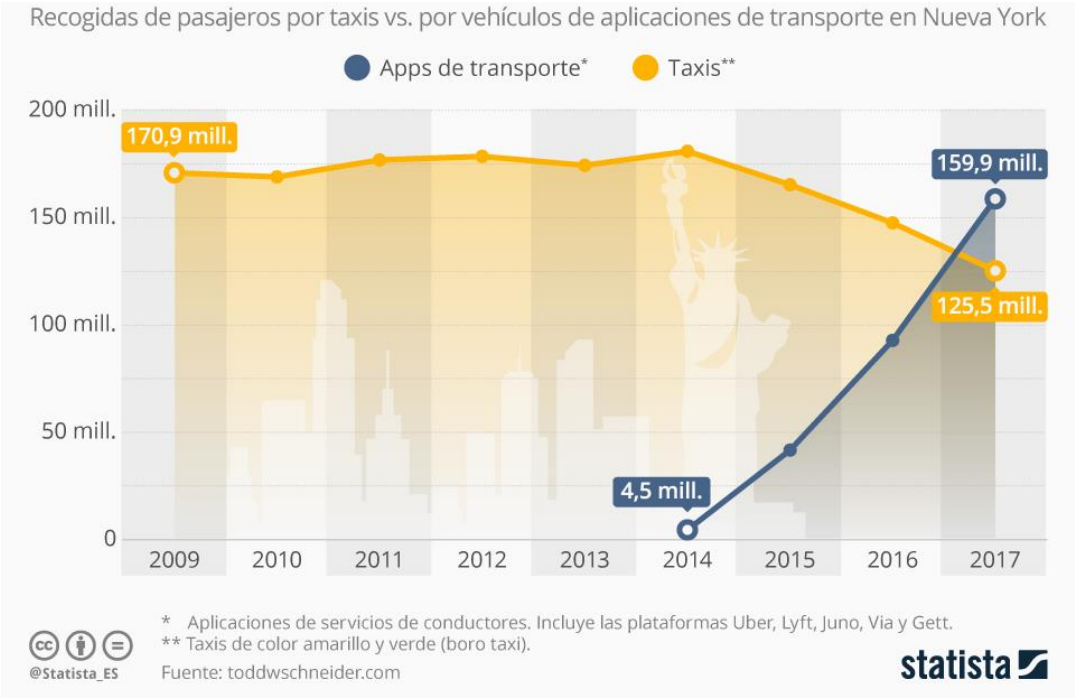
"La calidad del aire correspondiente al año 2019 en la ciudad de Madrid ha mejorado con respecto al año anterior en la mayoría de los contaminantes medidos en la red de vigilancia de la calidad del aire, aunque **esta mejoría no ha sido suficiente para el cumplimiento de los valores límite y umbrales establecidos para el dióxido de nitrógeno (NO2) y el ozono troposférico, que como en años anteriores se han superado**"

Fuente: Ayuntamiento de Madrid. "Memoria Anual Calidad del Aire 2019"

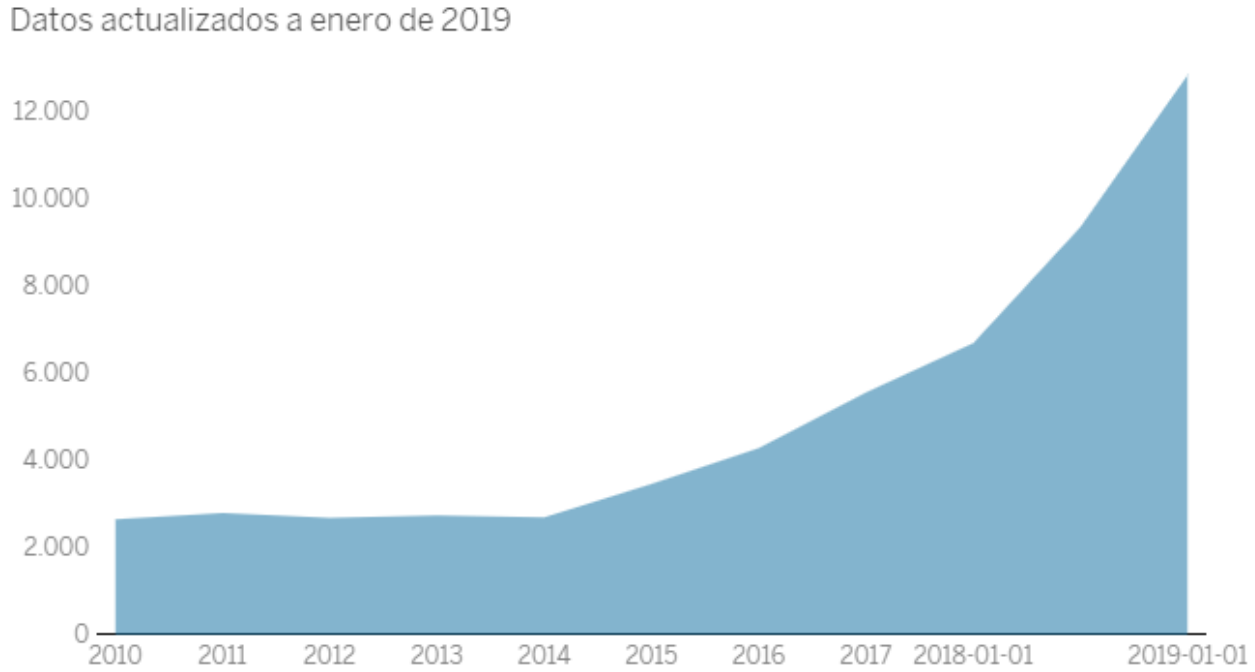
1 INTRODUCCIÓN

Las nuevas empresas basadas en aplicaciones móviles, como Uber y Cabify, han irrumpido con fuerza en los diferentes mercados locales

Recogidas de pasajeros por taxis frente a vehículos de aplicaciones móviles en Nueva York



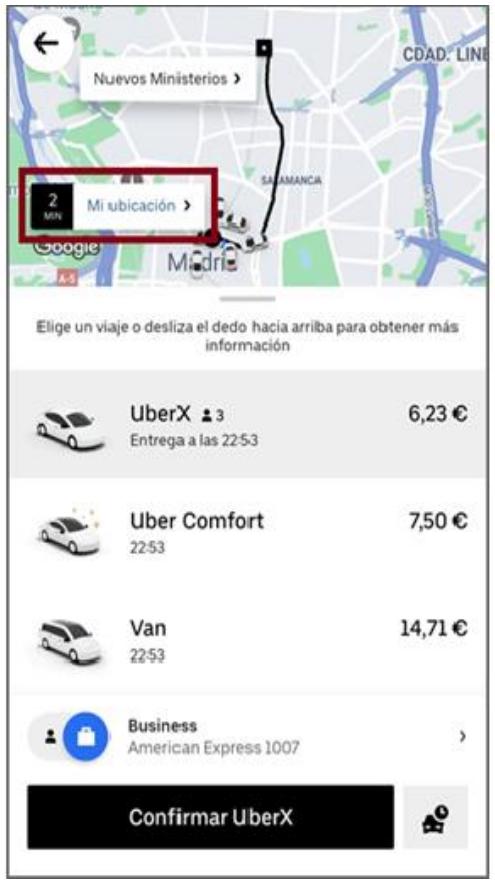
Evolución de las licencias VTC en España



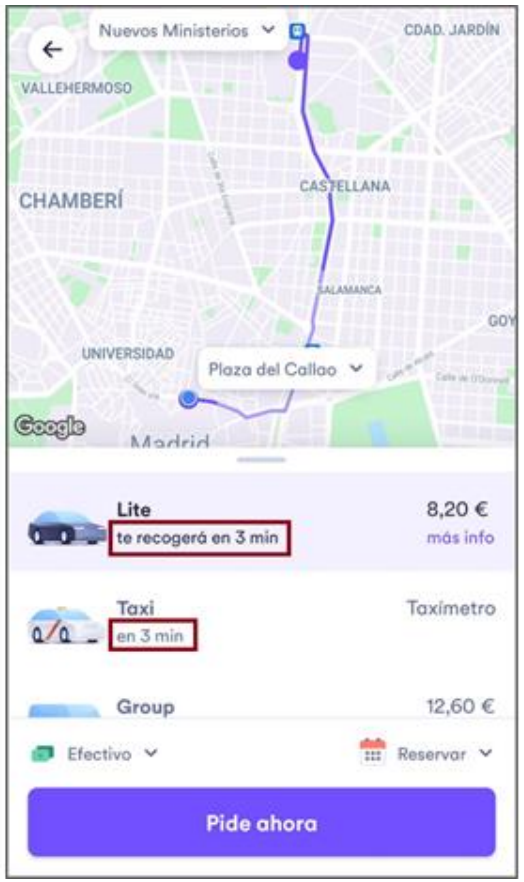
1 INTRODUCCIÓN

¿Qué ofrecen las principales **aplicaciones de movilidad (VTC)**?

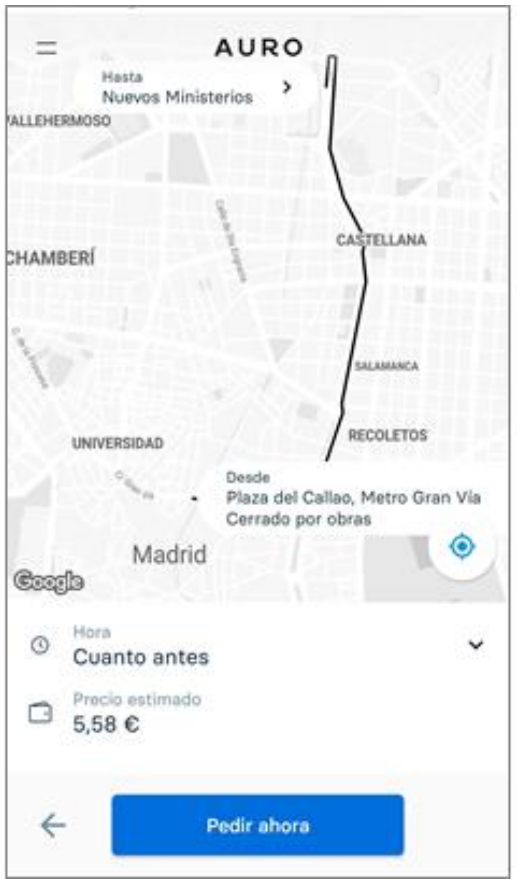
Uber



Cabify



Auro

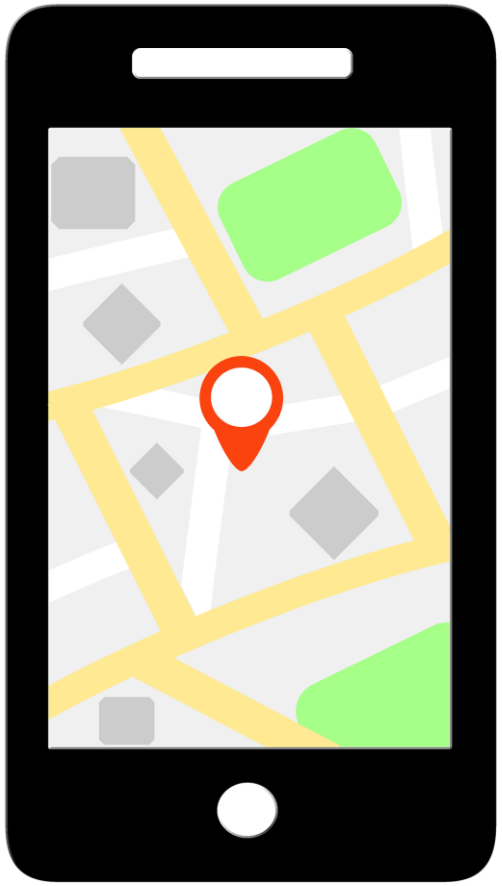


2

PROBLEMA PLANTEADO

2 PROBLEMA PLANTEADO

Mostrar una estimación del tiempo de recogida en la pantalla principal de una aplicación móvil



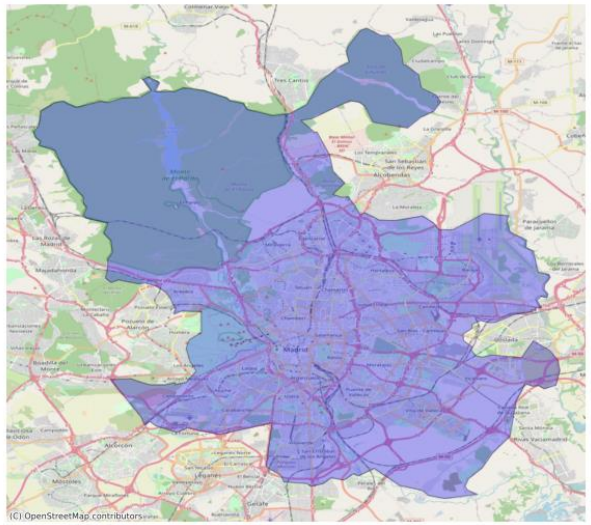
Las estimaciones devueltas deben ser fiables y ajustadas a la realidad

El tiempo de respuesta debe ser mínimo (ms), para garantizar una experiencia de usuario óptima

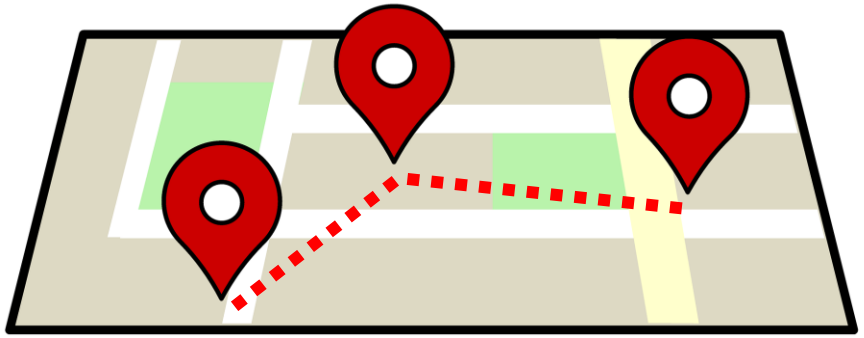
2 PROBLEMA PLANTEADO

Este problema puede dividirse en dos subtareas perfectamente diferenciadas

A **Predecir la posición del taxi** que recogerá al usuario



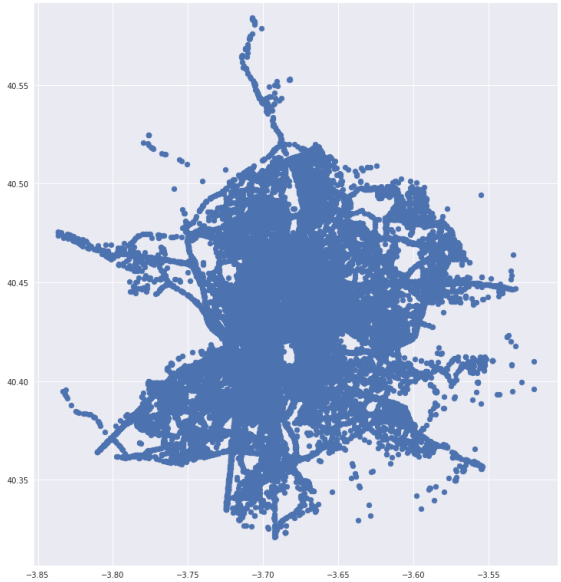
B **Estimar el tiempo de viaje** entre la posición del taxi y la ubicación de recogida del usuario



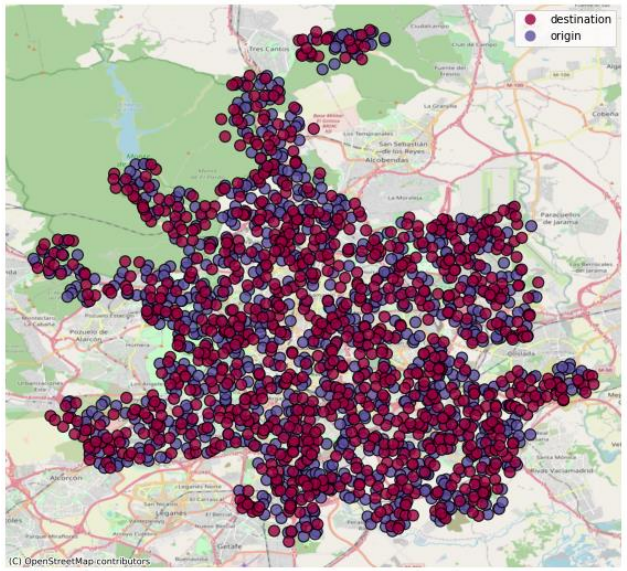
2 PROBLEMA PLANTEADO

Para cada uno de estas subtareas contamos con una fuente principal de información

A Histórico de la **disponibilidad de los taxis** en la ciudad de Madrid



B Histórico de **tiempos de viaje** entre varios puntos de Madrid



2 PROBLEMA PLANTEADO

Asociados a estas fuentes de datos algunas de las principales limitaciones son:

Los datos geográficos deben ser agregados para poder analizarse correctamente.

Es necesario definir áreas en las que agrupar estos datos.

Únicamente hay información de la disponibilidad de los taxis durante los últimos 3 meses.

Esto no permite capturar ciertos sesgos de estacionalidad.

Los viajes en taxi registrados tienen un volumen mínimo, 159 registros en el periodo de análisis.

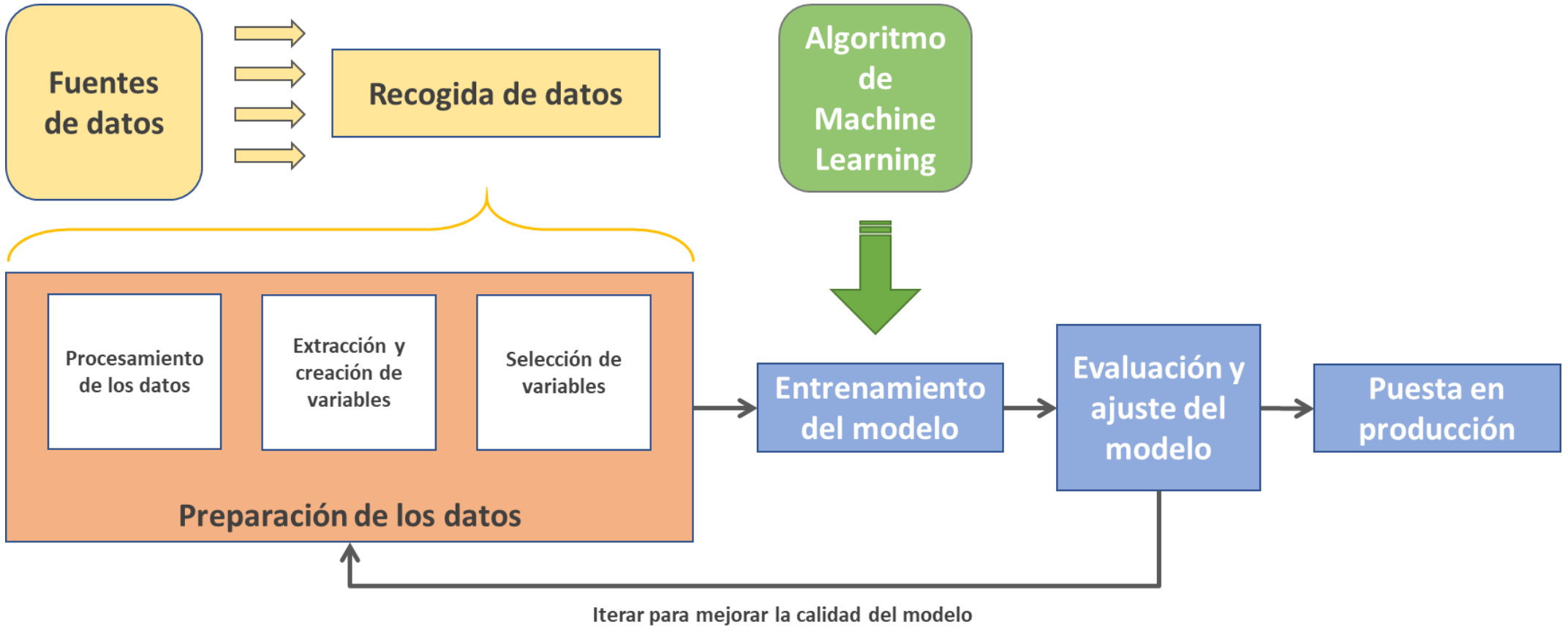
Es necesario identificar/construir una fuente de datos alternativa.

3

DISEÑO Y DESARROLLO

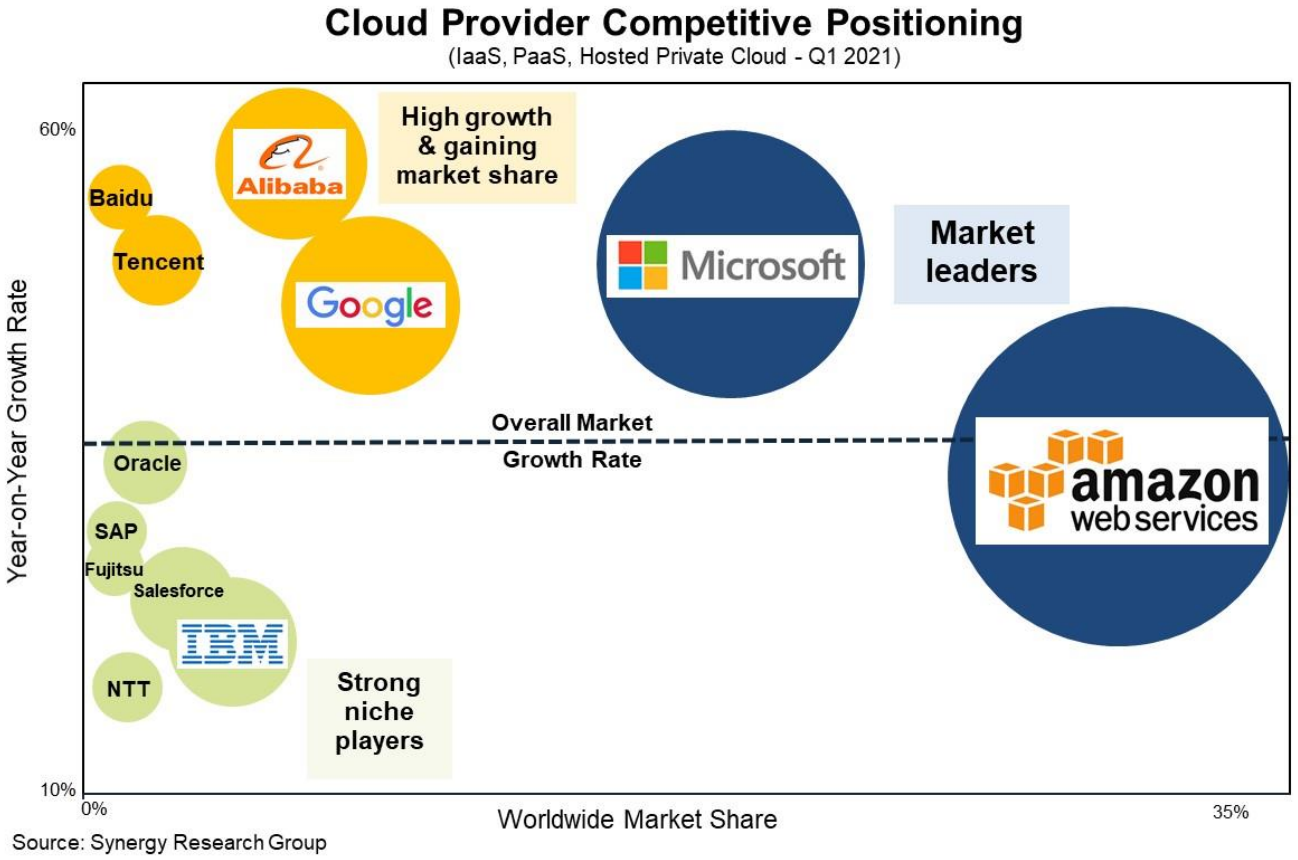
3 DISEÑO Y DESARROLLO

Para alcanzar el objetivo del proyecto, se ha definido un proceso con varias fases que abarcan desde la recogida de datos hasta la puesta en producción.



3 DISEÑO Y DESARROLLO

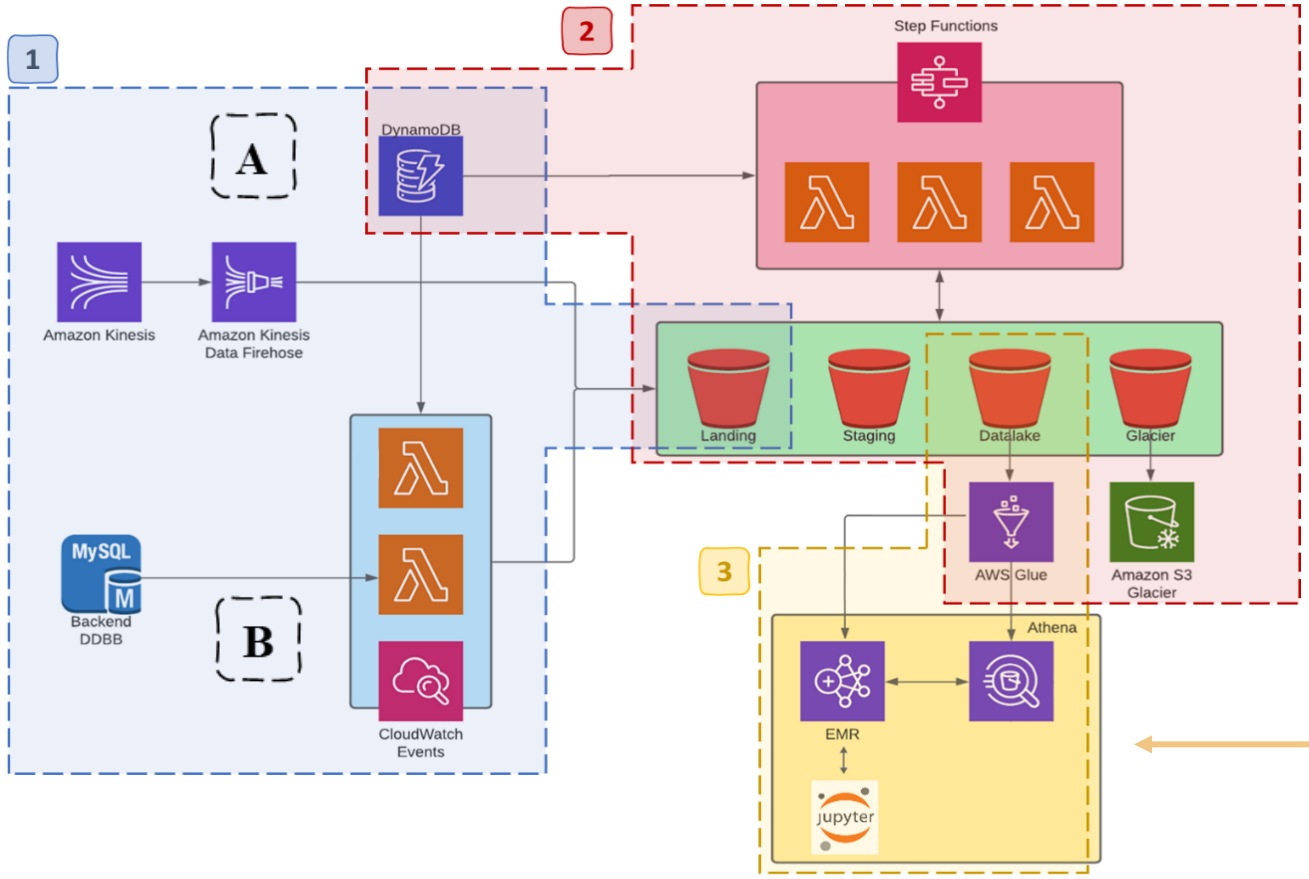
Los requisitos del proyecto determinan la necesidad de disponer de un sistema que ofrezca **alta disponibilidad**, **escalabilidad** y capacidad para la **recolección, tratamiento, almacenamiento y análisis de datos.**



3 DISEÑO Y DESARROLLO

La infraestructura para la realización del proyecto se ha implementado sobre la plataforma de *cloud* público de AWS.

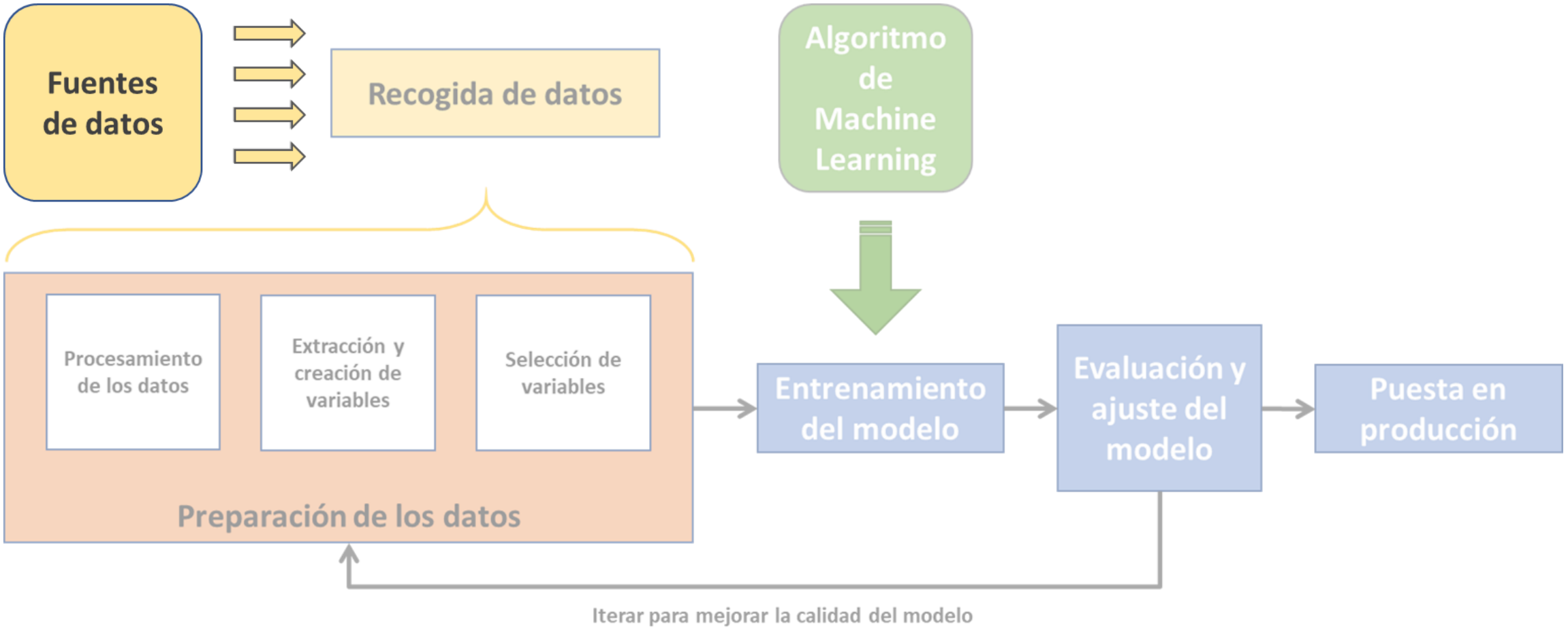
Ingestas de datos, “push” y “pull”, desde las fuentes de origen



Procesamiento de datos y actualización del catálogo

Análisis de datos y desarrollo de modelos predictivos

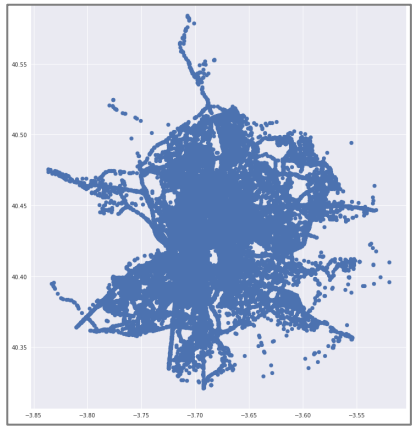
3 DISEÑO Y DESARROLLO



3 DISEÑO Y DESARROLLO

En el proyecto se han utilizado varias fuentes de datos (4). Tanto internas como externas.

Servicio de **taxis**, históricos de **disponibilidad y viajes realizados**



- ID
- FECHA
- CIUDAD
- LAT
- LON

- ID
- FECHA
- ORIGEN
- DESTINO
- ESTADO
- TIEMPO
- ETC.

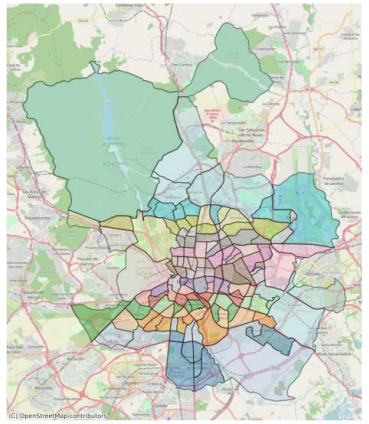
Histórico de trayectos entre dos puntos (API Google Maps)

- FECHA
- ORIGEN (LAT y LON)
- DESTINO (LAT y LON)
- DISTANCIA METROS
- DURACION SEGUNDOS
- DURACION SEGUNDOS TRAFICO
- ETC.

GADM - **Áreas administrativas de España** (CCAA, provincias, municipio)

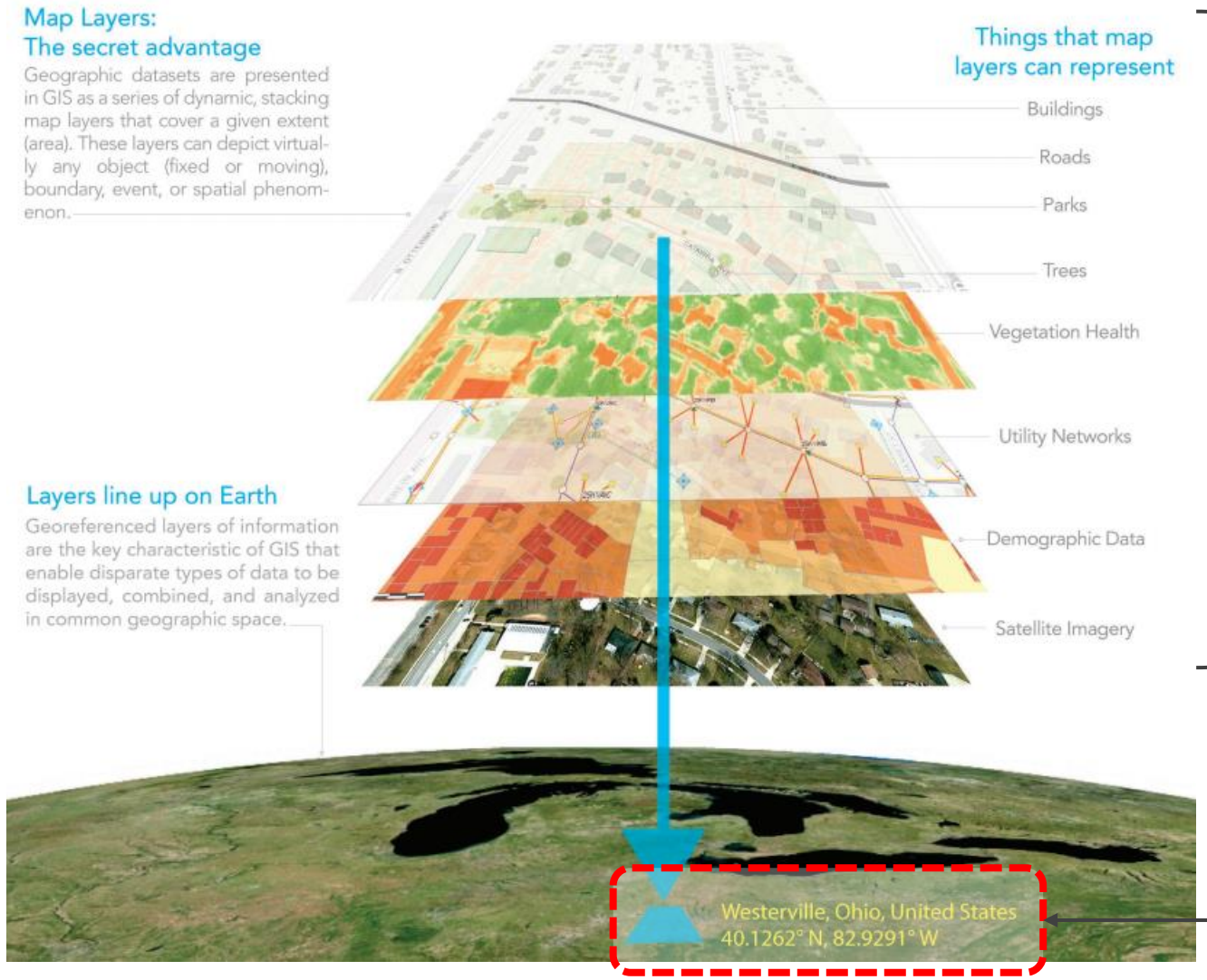
- PAIS
- COMUNIDAD AUTONOMA
- PROVINCIA
- REGION
- MUNICIPIO
- *GEOMETRY*

Áreas administrativas de Madrid (geoportal del Ayto.)



- ID OBJETO
- DISTRITO
- BARRIO
- *GEOMETRY*

3 DISEÑO Y DESARROLLO



Los datos geográficos hacen referencia a **localizaciones en la superficie terrestre**

GIS (*geographic information system*) **es el estándar** en este ámbito. Permite organizar mapas y datos en capas.

Este framework cuenta con **formatos** ya establecidos como **Shapfile (shp)** y otros más novedosos como **Geopackage (gpkg)**, entre otros

3 DISEÑO Y DESARROLLO

En función del tipo de fuente de datos podemos distinguir dos casuísticas diferenciadas.

Fuente de datos **PULL**

Los datos son extraídos de la fuente original mediante la llamada a una API o una consulta a la base de datos.

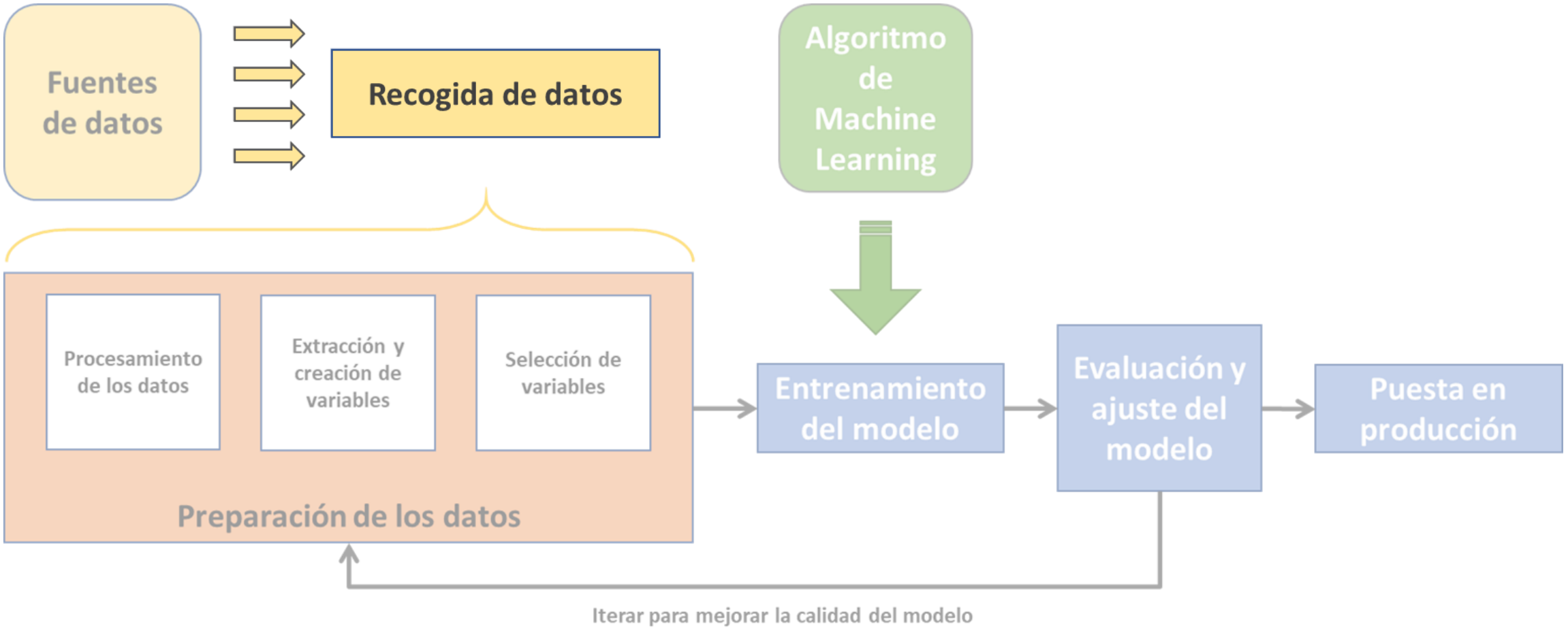
- Datos almacenados en bbdd (viajes en taxi)

Fuente de datos **PUSH**

Los datos se ingestan tan pronto como son generados y recibidos. Son recogidos en una cola de Kinesis.

- Datos recibidos desde el servicio de taxis (ubicación de los taxis disponibles)

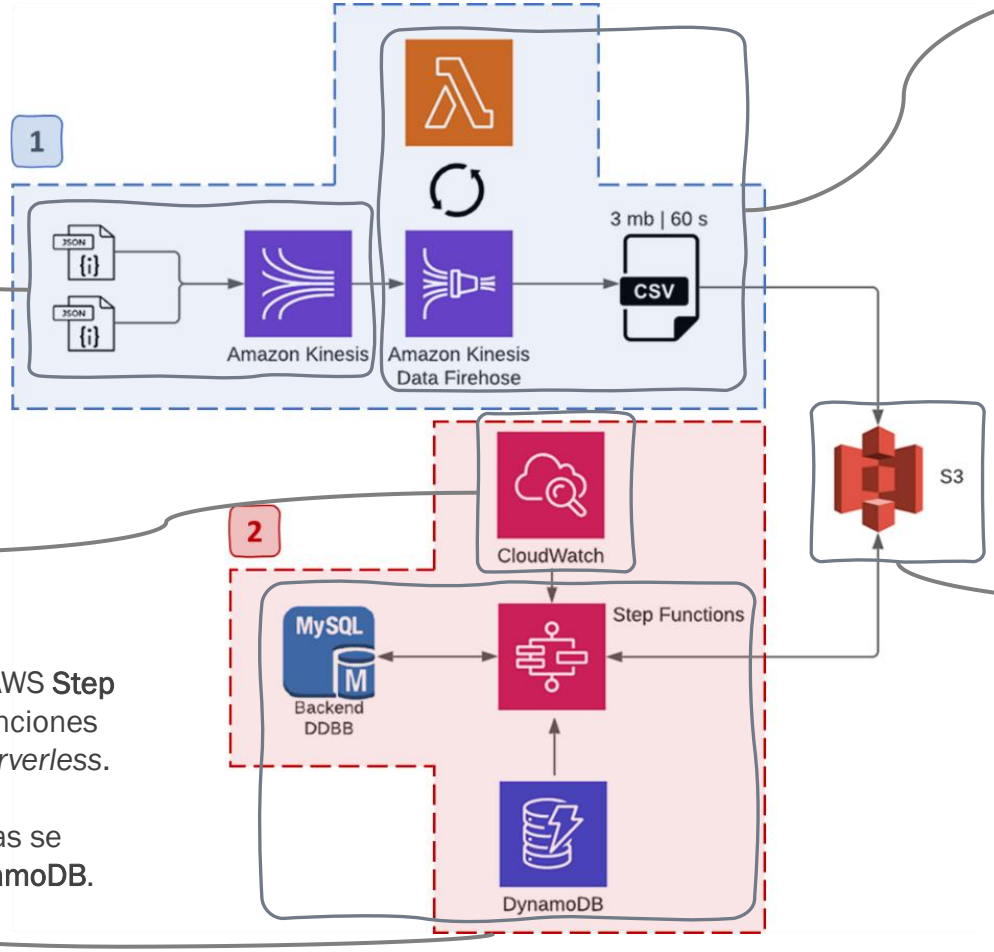
3 DISEÑO Y DESARROLLO



3 DISEÑO Y DESARROLLO

Existen dos procesos de ingesta diferenciados para estos tipos de ingesta (*pull* y *push*). Ambos convergen en el mismo destino.

Los **datos en tiempo real** son recogidos y procesados por AWS **Kinesis**. Estos datos se reciben en formado JSON.



Kinesis Data Firehose almacena los datos recibidos hasta alcanzar cierto tamaño o límite temporal antes de enviarlos a AWS S3. Además, los datos se transforman a formato CSV utilizando una función Lambda.

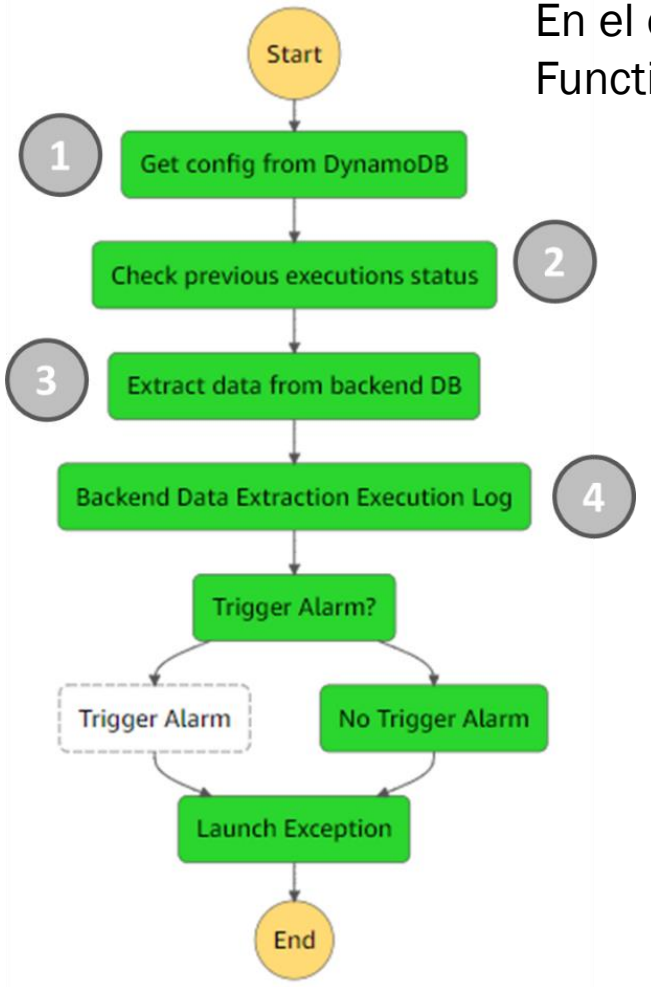
Los procesos de **ingesta *batch*** se disparan mediante eventos en AWS **Cloudwatch**. **Programados** en intervalos periódicos.

El flujo de la ingesta es **orquestado** mediante AWS **Step Functions** y los datos se procesan mediante funciones AWS **Lambda** en un arquitectura totalmente *serverless*.

Los **parámetros de configuración** de las ingestas se almacenan en una base de datos NoSQL, **DynamoDB**.

Finalmente, **los datos son almacenados** en un formato estructura (CSV) en el *bucket* "**Landing**" de AWS **S3**.

3 DISEÑO Y DESARROLLO



En el caso de las **ingestas *pull***, se han orquestado los procesos utilizando AWS Step Functions desde donde se invocan distintas funciones Lambda. Estos son los pasos:

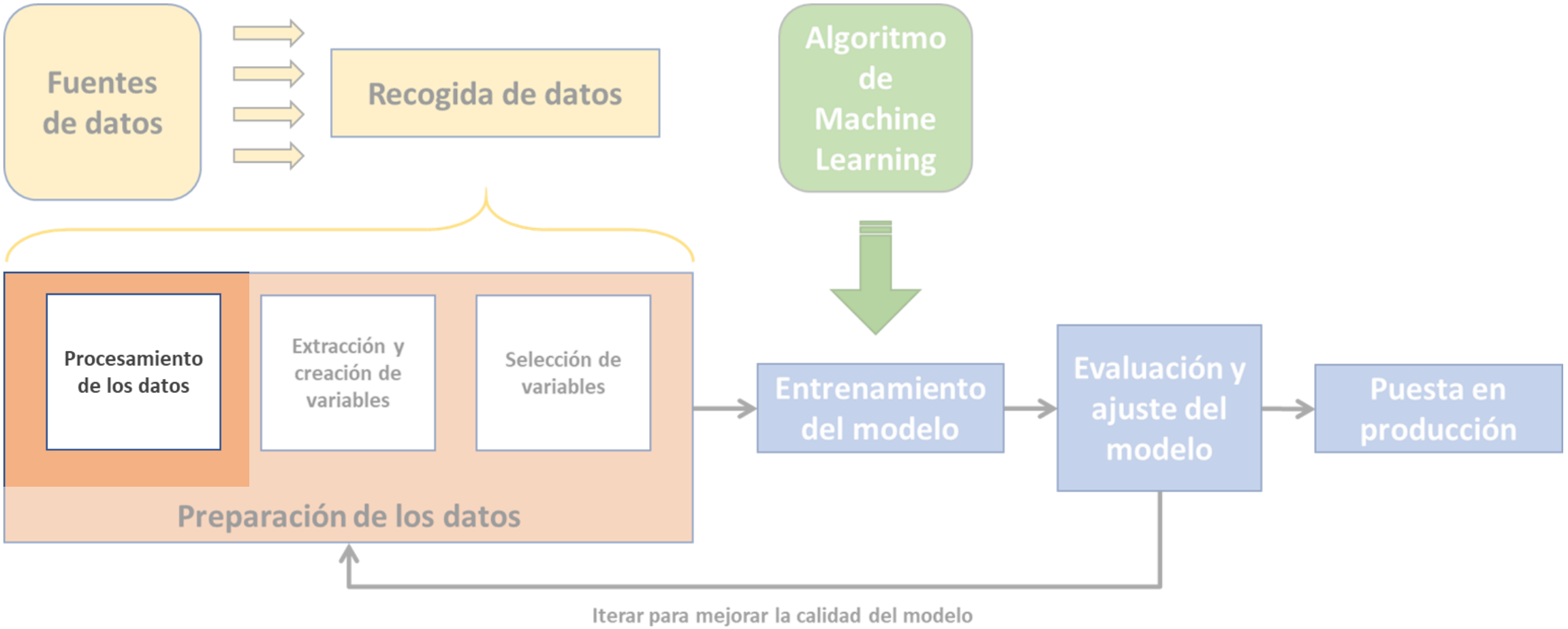
1. **Extraer los parámetros de configuración** (clave-valor) desde DynamoDB.
   Por ejemplo, rutas, detalles de la conexión con la BBDD, etc.

2. **Comprobar el estado de la ejecución previa** y definir los datos a extraer.

3. **Conexión con la BBDD MySQL y extracción de los datos** utilizando Python y sus conectores. Los datos se almacenan en un bucket de AWS S3.

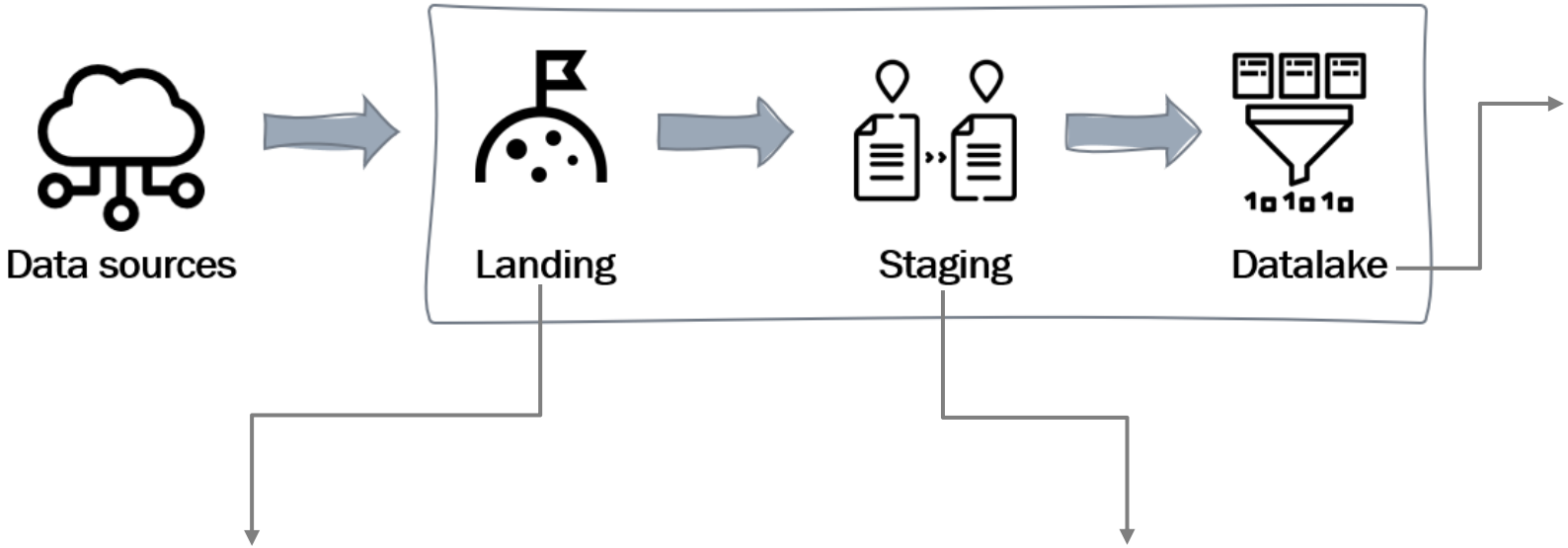
4. **Comprobar el resultado de la ejecución**, actualizar la tabla de estados con un nuevo registro. En caso de errores se dispara una alerta.

3 DISEÑO Y DESARROLLO



3 DISEÑO Y DESARROLLO

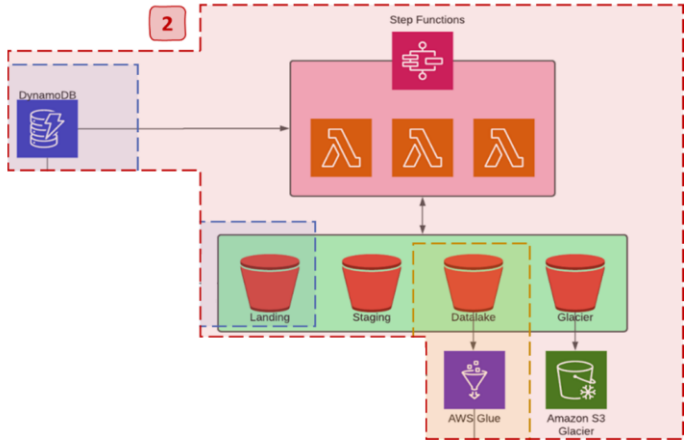
El procesamiento de los datos se ha dividido en 3 etapas desde la llegada del dato hasta el almacenamiento final. Cada una de ellas se localiza en un *bucket* diferente de AWS S3.



Finalmente, los datos procesados y transformados se almacenan en el "**Datalake**", en formato ***parquet***.

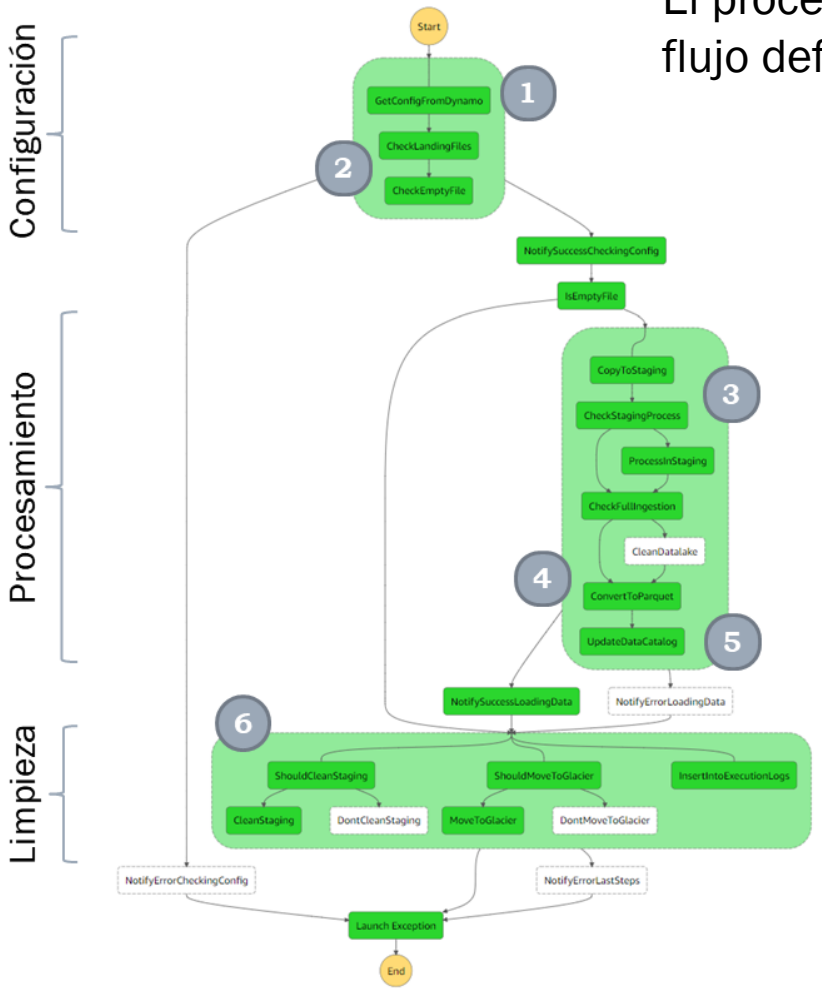
Los datos recogidos en las ingestas son almacenados en "***Landing***" inalterados, en formato ***csv***.

Estos datos son copiados a "**Staging**", un almacenamiento temporal donde se realizan diferentes transformaciones intermedias.

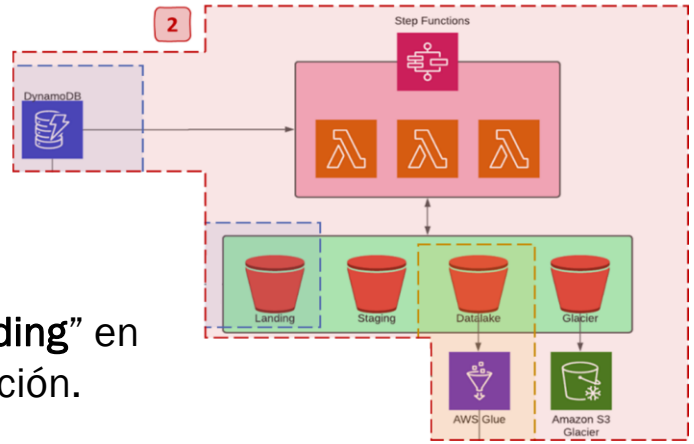


3 DISEÑO Y DESARROLLO

El procesamiento de los datos se ha orquestado nuevamente mediante un flujo definido en AWS Step Functions que invoca distintas funciones Lambda.



1. Extraer los **parámetros de configuración** desde DynamoDB.
2. Comprobar que los ficheros de "**Landing**" contienen **datos válidos**.
3. **Copiar** los datos a "**Staging**" **y procesar** la información (limpieza y normalización).
4. Convertir a **formato Parquet** y **almacenar** el dataframe en el **"Datalake".**
5. Actualizar los **metadatos** en el **catálogo** de datos de AWS Glue.
6. **Limpiar** "**Staging**", **archivar** los datos de "**Landing**" en "**Glacier**" y actualizar los registros de la ejecución.



3 DISEÑO Y DESARROLLO

Por último, una vez se han almacenado los datos procesados, se han llevado a cabo diferentes análisis, combinaciones y transformaciones de los mismos en un sistema analítico.

AWS S3 (DATALAKE)

Los datos procesados se almacenan en un bucket de AWS S3 en formato Parquet (un formato comprimido y columnar que almacena los datos eficientemente para realizar análisis sobre estos. Además incluye en el propio fichero los metadatos.

AWS GLUE DATA CATALOG

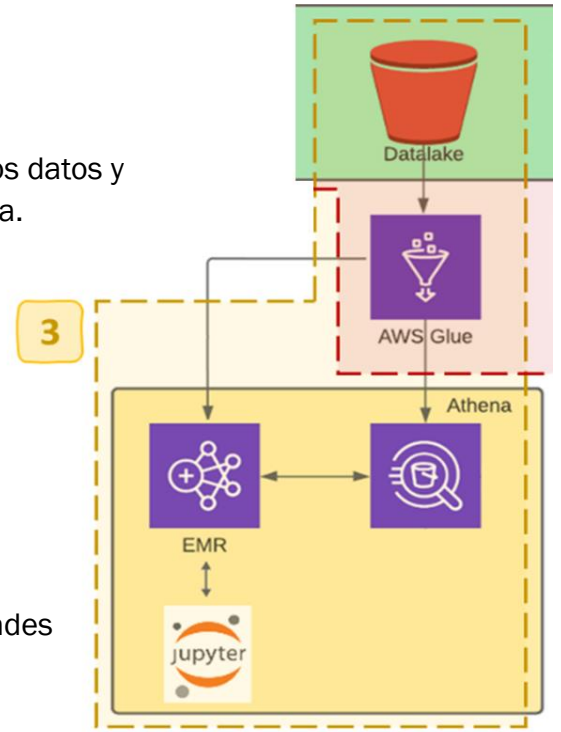
Glue es un repositorio unificado de metadatos que incluye un índice a la localización de la información, un esquema de los datos y métricas de ejecución. Es posible acceder a este catálogo desde numerosos servicios de AWS, por ejemplo, EMR o Athena.

AWS ATHENA

Athena es un servicio de AWS que permite realizar consultas para analizar los datos almacenados en AWS S3 utilizando un lenguaje SQL estándar.

AWS EMR + JUPYTERLAB *(On-Demand)*

EMR es un **clúster administrado** que permite ejecutar *frameworks* de Big Data, como Spark, para procesar y analizar grandes volúmenes de datos. Además, permite gestiona el despliegue de **Jupyter Hub**, donde es posible crear diferentes **Jupyter notebooks**. Esto son documentos que contienen código "*vivo*", ecuaciones, modelos analíticos, visualizaciones y texto.



3 DISEÑO Y DESARROLLO

Python, uno de los lenguajes más populares en el análisis de datos, ofrece numerosas librerías de código abierto el tratamiento de estos.

Tratamiento de datos

NumPy

pandas

Representación de datos

matplotlib

Folium

Procesamiento de datos geográficos

GeoPandas

UBER

Google Maps Platform

Machine learning

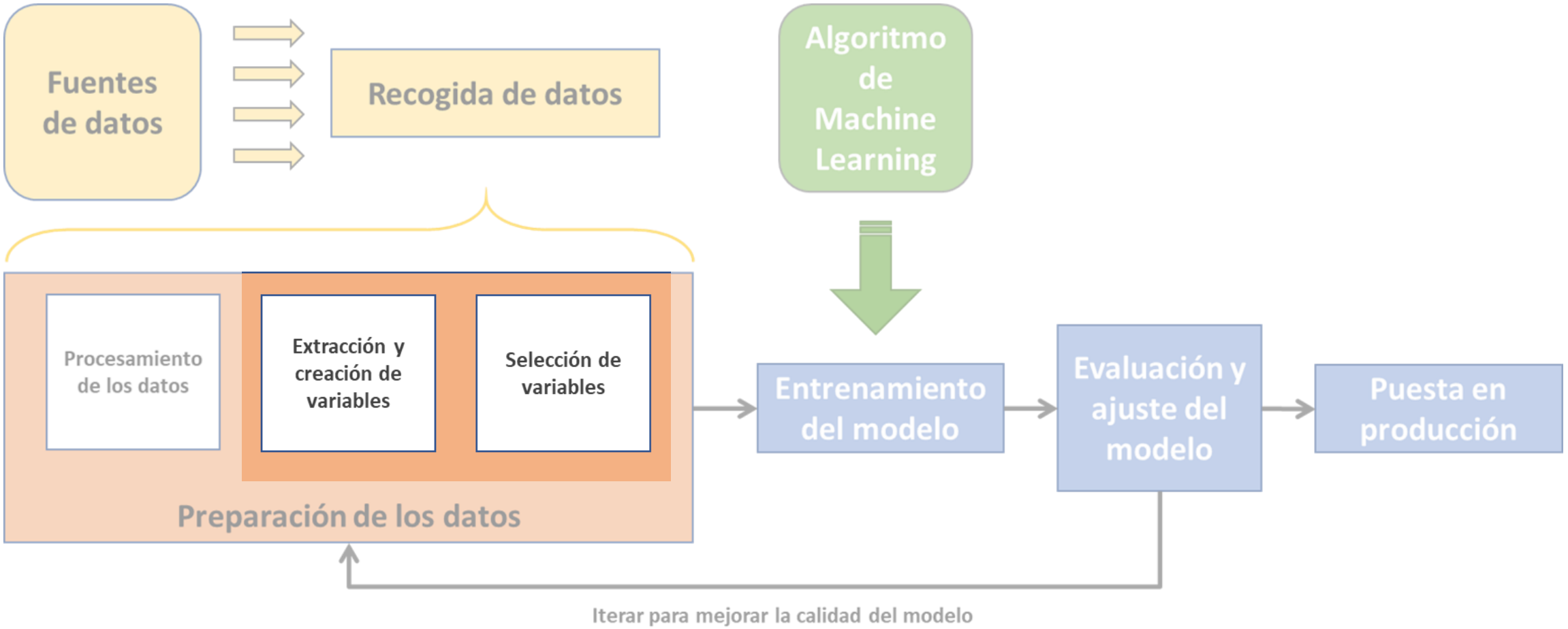
scikit learn

dmlc XGBoost

Integración con AWS

AWS Data Wrangler

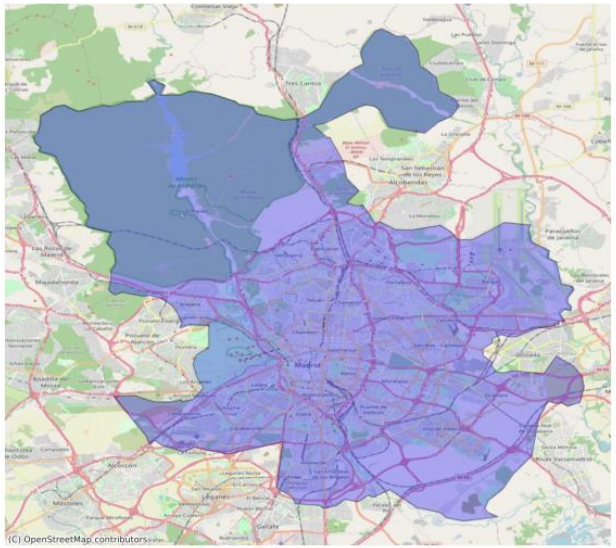
3 DISEÑO Y DESARROLLO



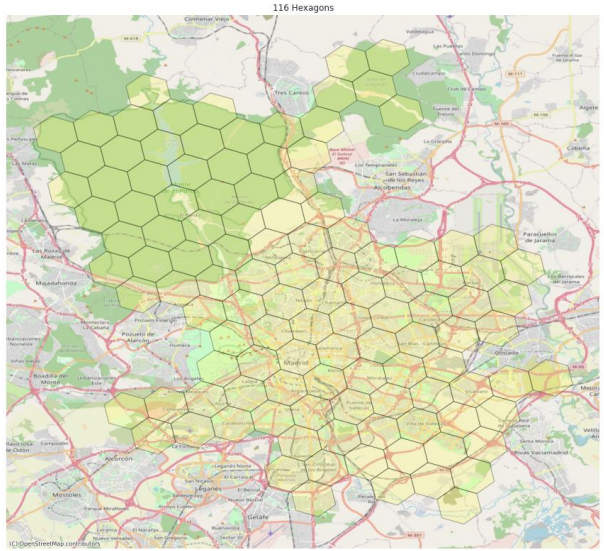
3 DISEÑO Y DESARROLLO

A partir de los datos procesados y almacenados en el *datalake*, se ha realizado el tratamiento de los mismos para enriquecer la información disponible mediante:

Definición de **los límites geográficos** del municipio de Madrid y **filtrado** de datos. Utilizando el *dataset* de **GADM**.



División del área metropolitana de Madrid en **áreas hexagonales**. Mediante la librería de Uber **H3**.

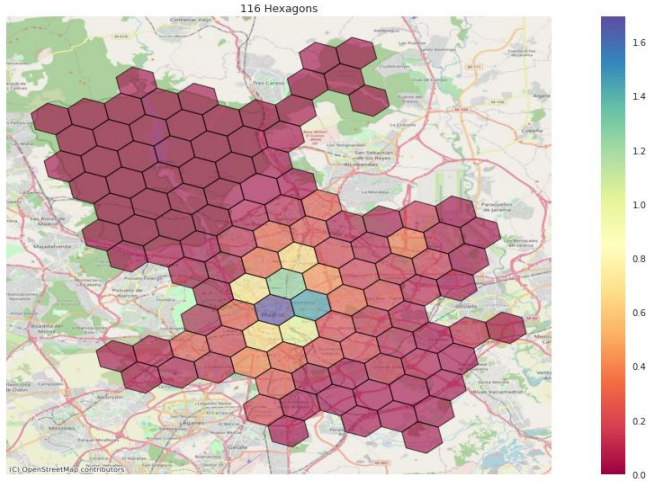


Enriquecimiento de los datos construyendo **nuevas variables** derivadas de las existentes, como:

- **Día de la semana**
- **Indicador de fin de semana**
- **Hora del día**
- **Periodo del día**
- **Parte de la hora**

3 DISEÑO Y DESARROLLO

Creación de una nueva fuente de datos, un histórico de trayectos entre dos puntos, mediante la API de Google Maps.



1

Debido al coste de cada llamada, existe una limitación de prepuesto que permite realizar **50.000 llamadas a la API**.

Por ello, es necesario definir una estrategia que permita **optimizar los pares origen-destino seleccionados**.

2

Para ello, se ha analizado la **tasa de disponibilidad histórica** en cada una de las **áreas** definidas.

A partir de estos datos, se han clasificado las áreas en cuatro grupos: **“hot”, “mild”, “cold” y “empty”**. Siendo las áreas “hot” las que mayor tasa de disponibilidad y las “empty” las de menor.

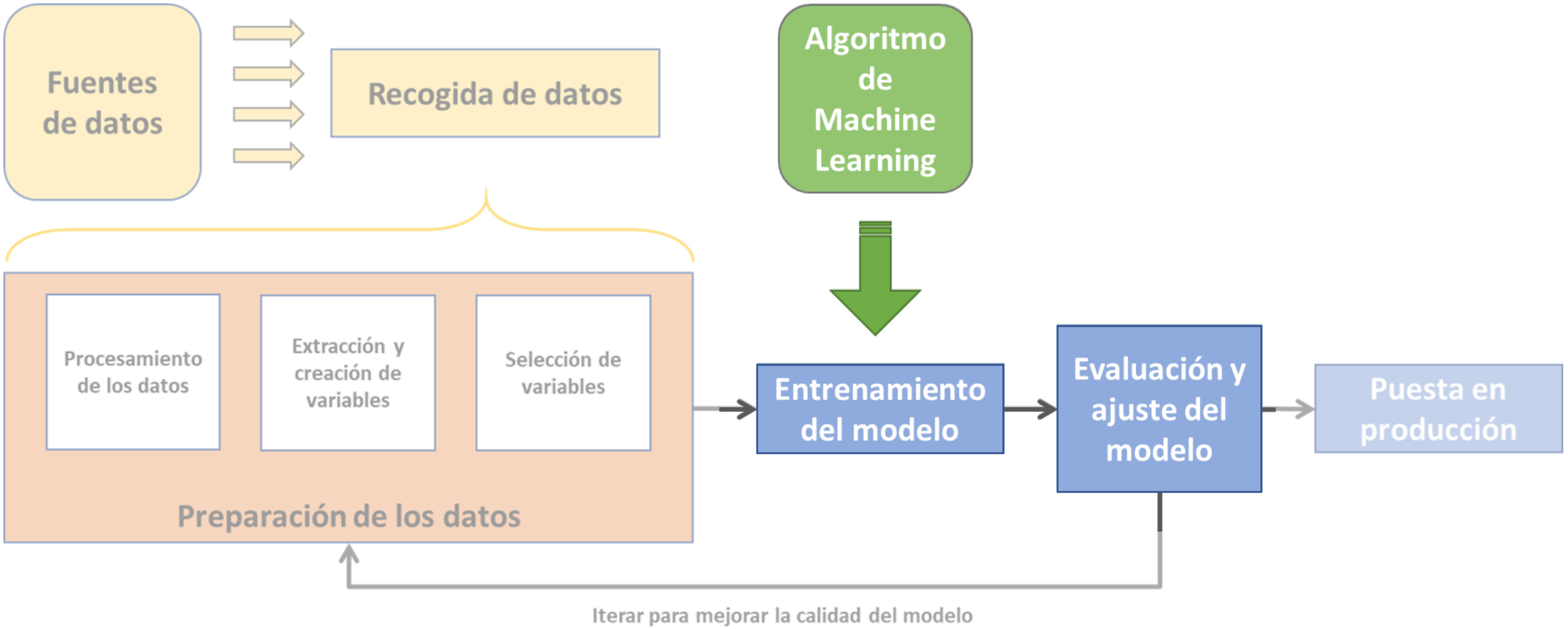
Origen	#Áreas	Destino	#Áreas	Repeticiones	Comentarios	Total
Cold	34	*Empty*	30	1	Petición única por área (1)	1.020
Cold	34	*Cold*	34	1	Petición única por área (1)	1.156
Mild	33	*Empty*	30	1	Petición única por área (1)	990
Mild	33	*Cold*	34	7	Una petición por cada día de la semana (7) y por área	7.854
Mild	33	*Mild*	33	7	Una petición por cada día de la semana (7) y por área	7.623
Hot	19	*Empty*	30	1	Petición única por área (1)	570
Hot	19	*Cold*	34	7	Una petición por cada día de la semana (7) y por área	4.522
Hot	19	*Mild*	30	28	Una petición por día de la semana (7), periodo del día (4) y área	15.960
Hot	19	*Hot*	19	28	Una petición por día de la semana (7), periodo del día (4) y área	10.108
TOTAL						49.803

3

A partir de las áreas definidas se han **priorizado aquellas de mayor tasa de disponibilidad**.

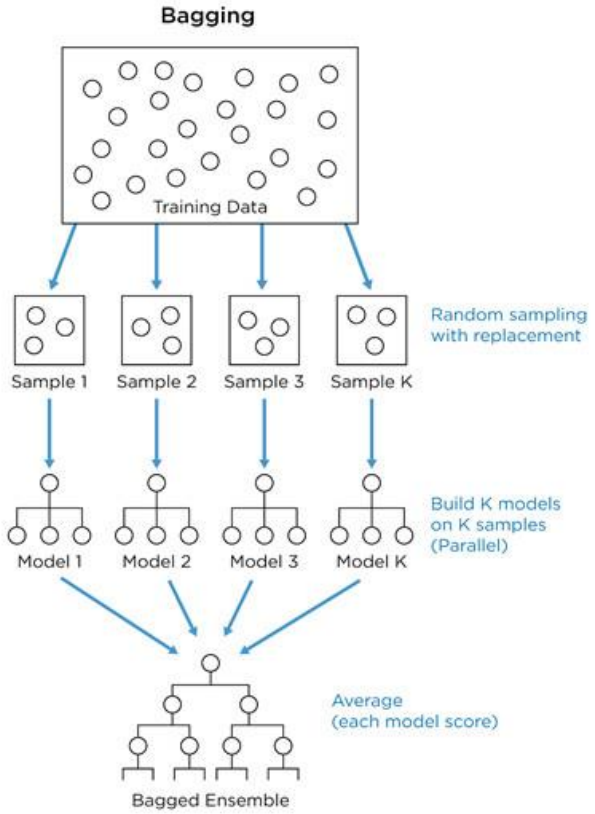
Es decir, los viajes desde áreas “hot” a otras áreas “hot” y a las “mild” tendrá asignado un alto número de llamadas. Por otro lado, desde las áreas “cold” al resto de áreas se asignara un número mínimo de llamadas.

3 DISEÑO Y DESARROLLO



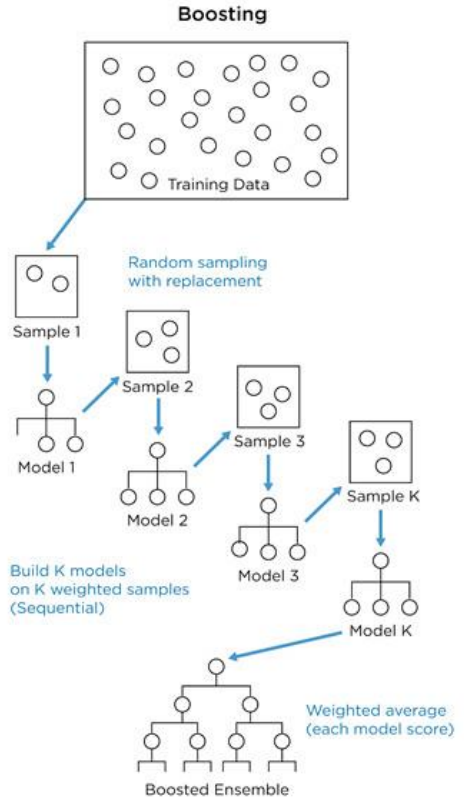
3 DISEÑO Y DESARROLLO

Los algoritmos de *ensemble learning* combinan diferentes modelos básicos en otros más complejos que mejoran el rendimiento global, produciendo resultados más precisos y reduciendo la tasa de error en numerosos escenarios.



En este proyecto:

Random Forest



En este proyecto:

Gradient Boosting

3 DISEÑO Y DESARROLLO

En primer lugar se ha elaborado un **modelo capaz de predecir el punto de partida del taxi** que recogerá al usuario, en un **momento temporal y ubicación definidos**, partiendo de los datos históricos de disponibilidad.

1 Partiendo del *dataset* con los **históricos de disponibilidad**, se ha filtrado los registros ubicados fuera del área metropolitana de Madrid (18%).

Manteniendo un total de **227.294 registros**.

2 Se ha **agregado** la información a nivel de área, día de la semana, periodo del día, hora del día y parte de la hora.

Tomando como variable objetivo la media calculada de taxis (únicos).

3 Únicamente se dispone de información de las áreas con taxis disponibles.

Es necesario completar la información con las **combinaciones de áreas y momentos temporales donde no había ningún taxi disponible**.

Como resultado, tenemos un conjunto de datos que cuenta con un total de **77.952 de registros**

4 Se han **definido** las ***features*** que entrarán al modelo y la **variable objetivo.**

5 Se han **transformado las variables categóricas**, **discretizándolas** en variables en numéricas.

6 Se han divido los registros del *dataset* de entrada en dos conjuntos: **training (70%)** y **test (30%)**.

3 DISEÑO Y DESARROLLO

A partir de estos datos se han entrenado diversos modelos aplicando un algoritmo de ***Random Forest***.

Para entrenar estos modelos de random forest, se ha utilizado la librería de Python "**SKLearn**".

Esta librería permite definir distintos **parámetros de configuración** que determinan la forma en que se entrenará el modelo.

Para obtener la **configuración óptima** que permita entrenar el modelo con mayor precisión y menor tasa de errores, se ha aplicado una técnica de "**tuning**" mediante la función "**Grid Search**".

Se han entrenado **135 modelos**

	oob_r2	max_depth	max_features	min_samples_split	n_estimators
1	0.930408	NaN	auto	2	250
0	0.930007	NaN	auto	2	100
4	0.775136	NaN	sqrt	2	100
8	0.775136	NaN	log2	2	100
9	0.770696	NaN	log2	2	250

Métrica a optimizar, coeficiente de determinación (R^2).

Indica la bondad del modelo y muestra si las variables independientes seleccionadas explican la variabilidad.

Profundidad máxima que pueden alcanzar los árboles.

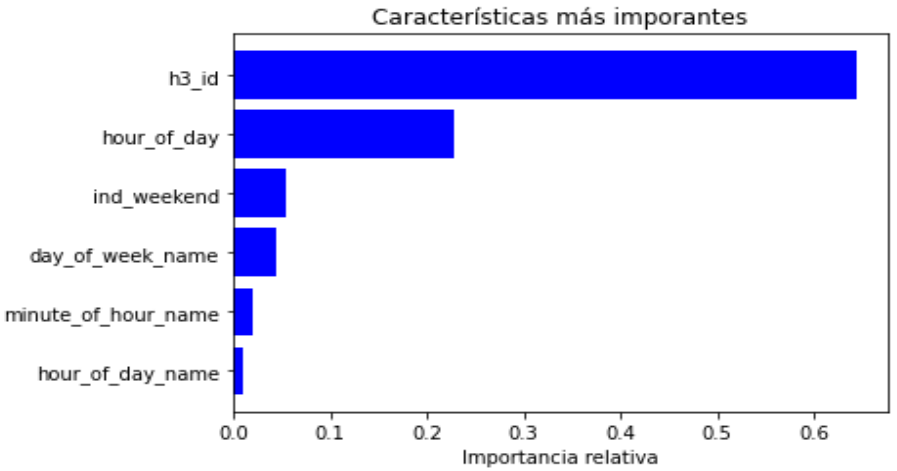
Número **máximo de predictores** considerados en cada división.

Número **mínimo de observaciones** de un nodo para poder dividirse.

Número de árboles incluidos en el modelo

3 DISEÑO Y DESARROLLO

Finalmente, se ha seleccionado la configuración con resultados óptimos: **250 estimadores, sin limitar ni el número de *features* ni la profundidad del árbol** y estableciendo un **mínimo de dos observaciones para que un nodo pueda dividirse.**



Para este modelo se han identificado las ***features* que más influyen en el resultado**, ordenados por su importancia relativa en el modelo:

- Área de recogida del usuario
- Hora del día
- Indicador de fin de semana
- Día de la semana
- Parte de la hora (en grupos de 15 minutos)
- Periodo del día (mañana → noche)

Se ha aplicado el modelo a una matriz con todas las combinaciones y se ha definido una "**categoría de disponibilidad**".

- Tasa de disponibilidad > 0.75, categoría de disponibilidad = 100
- Tasa de disponibilidad > 0.60, categoría de disponibilidad = 50
- Tasa de disponibilidad > 0.50, categoría de disponibilidad = 25
- Tasa de disponibilidad > 0.25, categoría de disponibilidad = 10
- Tasa de disponibilidad <= 0.25, estos elementos han sido descartados.

Por un lado, se ha calculado un score basado en la tasa de disponibilidad:

$$\boldsymbol{Score\_disp = 100\ x\ factor\_disponibilidad\ x\ categoria\_disponibilidad}$$

Por otro lado, se ha calculado otro score asociada a la distancia:

$$\boldsymbol{Score\_dist = 100\ x\ factor\_distancia/(distancia\_relativa + 1)}$$

Finalmente, se ha identificado el área de origen del taxi **combinando ambos *scorings*** con un peso del 50% en cada uno.

3 DISEÑO Y DESARROLLO

En segundo lugar, se ha elaborado un **modelo capaz de estimar el tiempo de viaje entre un origen y destino**, en un **momento temporal dado.** Como punto de partida se ha utilizado el histórico de viajes calculado con la API de Google.

1 Se ha construido un **histórico de viajes** entre diferentes puntos de Madrid y en diferentes momentos temporales, utilizando la API de Google Maps.

Este *dataset* cuenta con un total de **49.803 registros**.

2 Se han creado **campos adicionales** para enriquecer la información disponible, incluyendo: **variables temporales** (día de la semana, periodo del día, etc.), **distancia entre origen y destino** (en km y en nº de áreas intermedias), **indicador de vecinos** (origen-destino), etc.

3 Se ha llevado a cabo un **análisis, tratamiento y limpieza** de los datos disponibles:

- Análisis y tratamiento de ***missings***.
- Excluir las variables con **varianza próxima a cero**.
- **Estandarizar la escala** de variables numéricas.

4 Se han **definido** las ***features*** que entrarán al modelo y la **variable objetivo.**

5 Se han **transformado las variables categóricas**, **discretizándolas** en variables en numéricas.

6 Se han divido los registros del *dataset* de entrada en dos conjuntos: **training (70%)** y **test (30%)**.

3 DISEÑO Y DESARROLLO

A partir de estos datos se han entrenado diversos modelos aplicando un algoritmo de ***Gradient Boosting***.

Para entrenar estos modelos de ***gradient boosting***, se ha utilizado la librería de Python "**XGBoost**".

Esta librería permite definir numerosos **parámetros de configuración** que determinan la forma en que se entrenará el modelo.

Para obtener la **configuración óptima** que permita entrenar el modelo con mayor precisión y menor tasa de errores, se ha aplicado una técnica de "**tuning**" mediante la función "**Grid Search**".

	param_booster	param_learning_rate	param_max_depth	param_subsample	mean_test_score	std_test_score	mean_train_score	std_train_score
17	gbtree	0.1	None	1	-91.973870	0.480129	-64.285822	0.706910
23	gbtree	0.1	10	1	-92.269562	0.424406	-31.437946	3.270415
16	gbtree	0.1	None	0.5	-93.461750	0.517370	-63.478279	1.365415
22	gbtree	0.1	10	0.5	-94.206244	0.255947	-36.447036	3.976339
20	gbtree	0.1	5	0.5	-99.641649	0.640304	-79.054711	0.533393
21	gbtree	0.1	5	1	-101.591654	0.932667	-83.940655	0.846982
14	gbtree	0.01	10	0.5	-110.629547	0.794638	-79.933263	0.079942
15	gbtree	0.01	10	1	-113.678738	0.977849	-80.853208	0.630393
8	gbtree	0.01	None	0.5	-178.714809	1.551560	-169.360388	0.132197
9	gbtree	0.01	None	1	-187.555597	2.718895	-178.196529	1.326409

Indica el **tipo de refuerzo** a utilizar entre las diferentes posibilidades: gbtree, gblinear o dart.

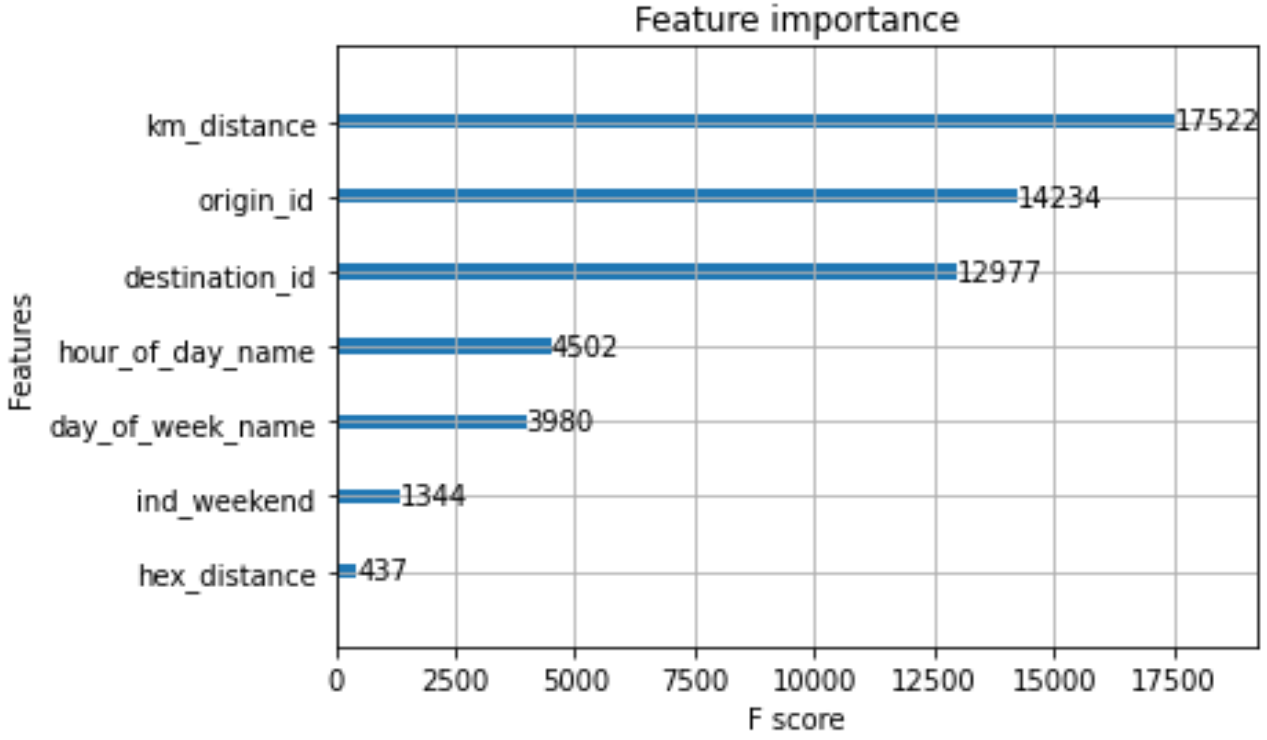
Tasa de aprendizaje. Reduce la contribución de cada árbol.

Profundidad máxima de los árboles.

Proporción de observaciones utilizadas en el ajuste de cada árbol.

3 DISEÑO Y DESARROLLO

A partir del análisis anterior, se ha seleccionado la configuración con resultados óptimos: **gbtree** como tipo de refuerzo, una **tasa de aprendizaje de 0.1**, **sin profundidad máxima** y seleccionando **todas las observaciones** en el ajuste.



Para este modelo se han identificado las ***features* que más influyen en el resultado**, ordenados por su importancia relativa en el modelo:

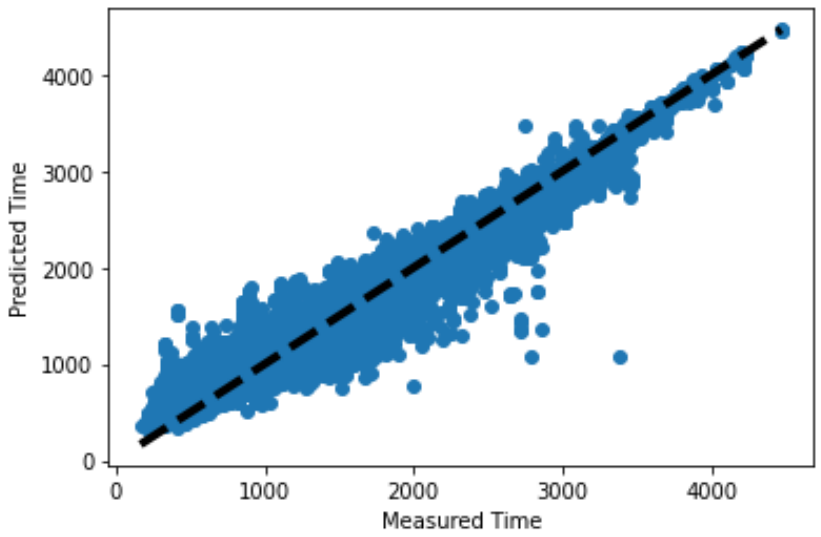
- Distancia en línea recta (en km) entre el origen y el destino.
- Área de origen del trayecto.
- Área de destino del trayecto.
- Periodo del día.
- Día de la semana.
- Indicador de fin de semana.
- Distancia entre el área de origen y destino, en base al número de áreas intermedias.

3 DISEÑO Y DESARROLLO

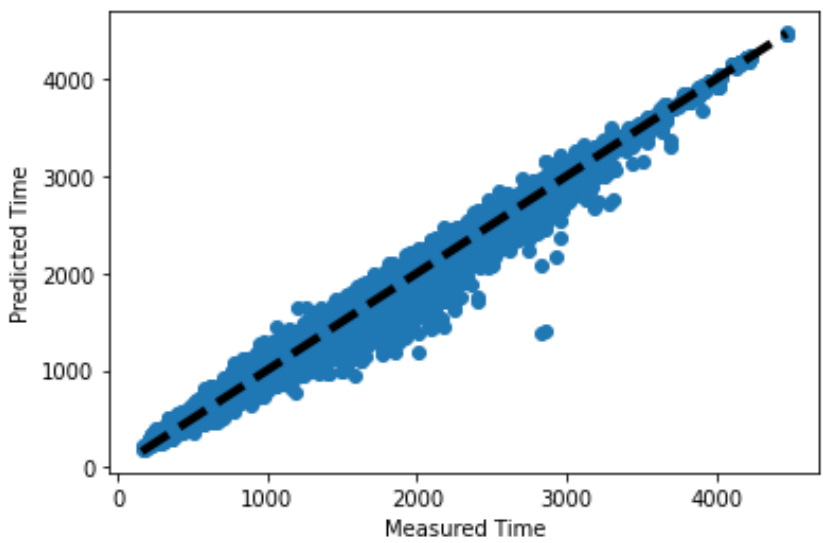
Por último, se han medido los resultados obtenidos por el modelo con el objetivo de validar los resultados del mismo.

En primer lugar, se han comparado los resultados obtenidos con el **modelo optimizado (2)** frente al **modelo inicial (1):**

1



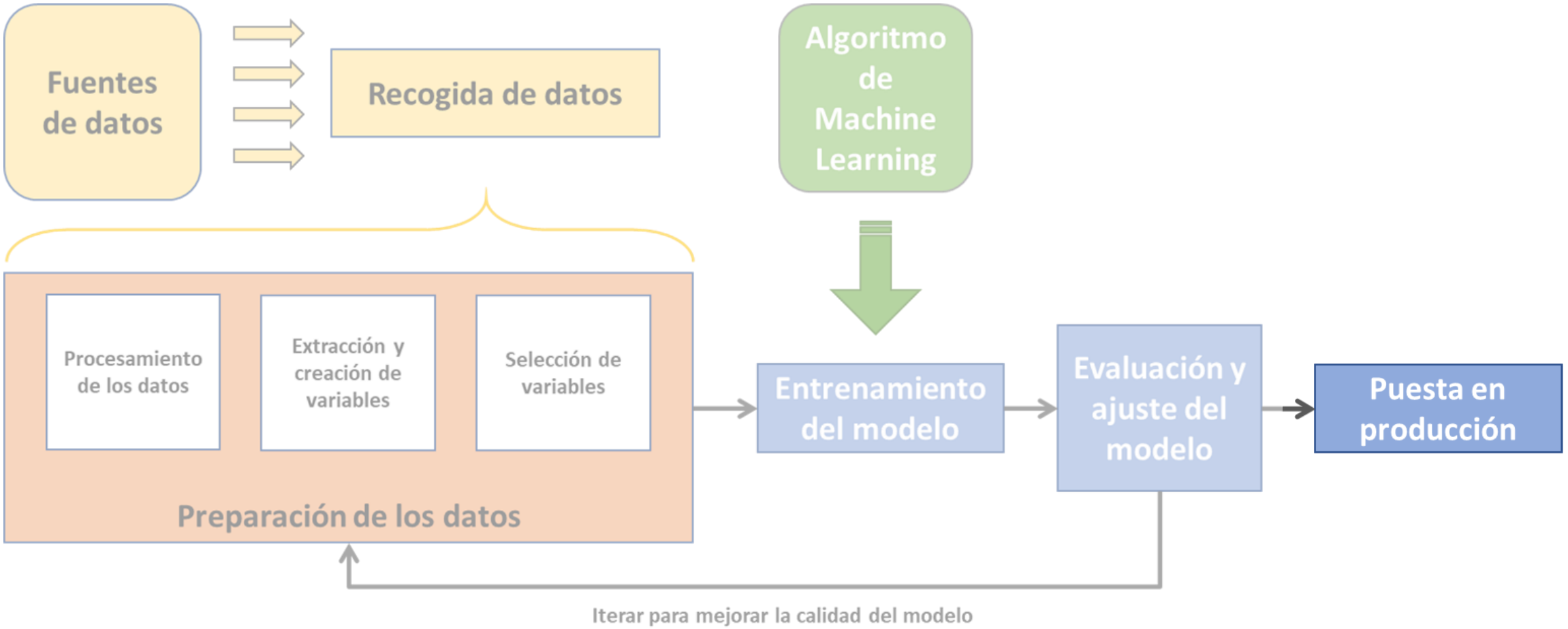
2



Comparando estos resultados se puede validar la mejora en los resultados obtenidos.

(*) Adicionalmente, se han calculado otras métricas (incluidas en la memoria) como RMSE, MAE o R^2 para evaluar este modelo.

3 DISEÑO Y DESARROLLO



3 DISEÑO Y DESARROLLO

Finalmente, se ha creado una **matriz con todas las posibles combinaciones** (área de recogida del usuario y fecha) y **se ha calculado el tiempo de espera** para cada registro a partir de los **dos modelos** desarrollados .

- **Áreas de recogida**, es decir, la ubicación de recogida del usuario (116).
- **Fecha del viaje**, día de la semana (7, incluido si es fin de semana o no).
- **Fecha del viaje**, hora del día (24, incluye los 4 periodos del día).
- **Fecha del viaje**, parte de la hora (4, minutos agrupados en franjas de 15 minutos).

Matriz con
77.952 registros

Sobre esta matriz, se ha aplicado el primer modelo para predecir el área de origen donde se encontrará el taxi y el segundo modelo para estimar el tiempo de espera.

Es decir, **todos los valores estarán precalculados** para ahorrar el tiempo de procesamiento necesario cuando se reciba una petición y reducir el tiempo de respuesta al mínimo.

4

INTEGRACIÓN Y RESULTADOS

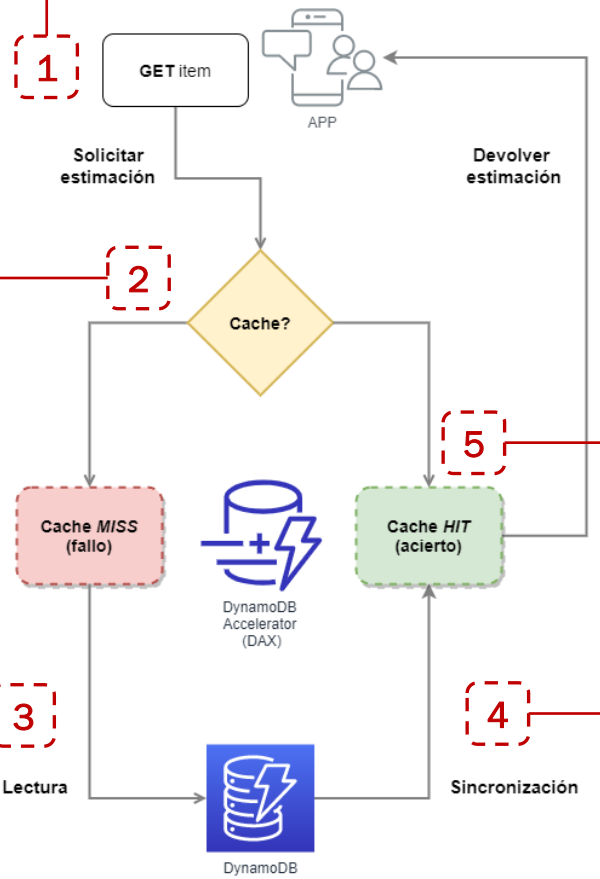
4 INTEGRACIÓN Y RESULTADOS

Para poner este sistema en producción e integrarlo con la APP es necesario definir un **sistema capaz de devolver las estimaciones en un tiempo de respuesta mínimo**. Se ha utilizado **DynamoDB y DAX**.

(1) La aplicación solicita una estimación.

(2) Se comprueba si dicho elemento se encuentra en la caché (DAX).

(3) Si el elemento no está presente en la caché (DAX), se busca en la base de datos (DynamoDB)



(6) Se devuelve la estimación a la aplicación móvil.

(5) Si el elemento está presente, se produce un acierto o cache hit y se devuelve el ítem .

(5) Se sincroniza la caché, para que este elemento esté disponible en la próxima lectura.

4 INTEGRACIÓN Y RESULTADOS

En primer lugar, se ha realizado un análisis de las diferentes zonas en función de la disponibilidad histórica de taxis.

Áreas HOT

Son áreas con una elevada tasa de disponibilidad de taxis.

Principalmente abarcan la almendra central y el aeropuerto.

Áreas MILD

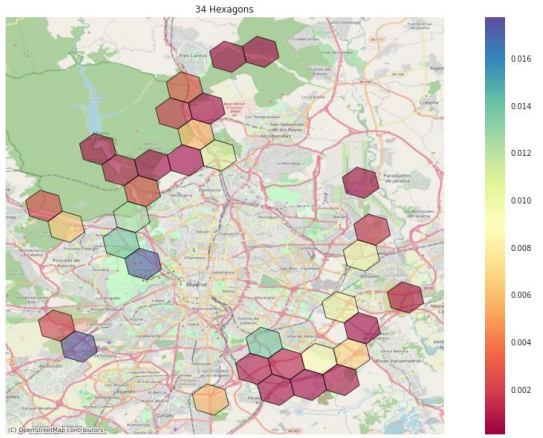
Son áreas con una tasa media de disponibilidad de taxis.

Principalmente abarcan las zonas que rodean la almendra central.

Áreas COLD

Son áreas con una baja tasa de disponibilidad de taxis.

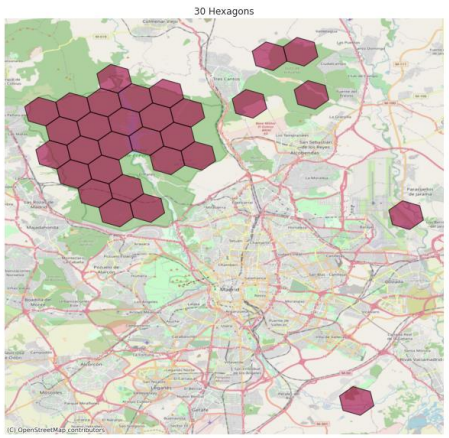
Principalmente abarcan zonas de la periferia, especialmente áreas poco pobladas.



Áreas EMPTY

Son áreas sin la presencia de ningún taxi en toda la serie histórica.

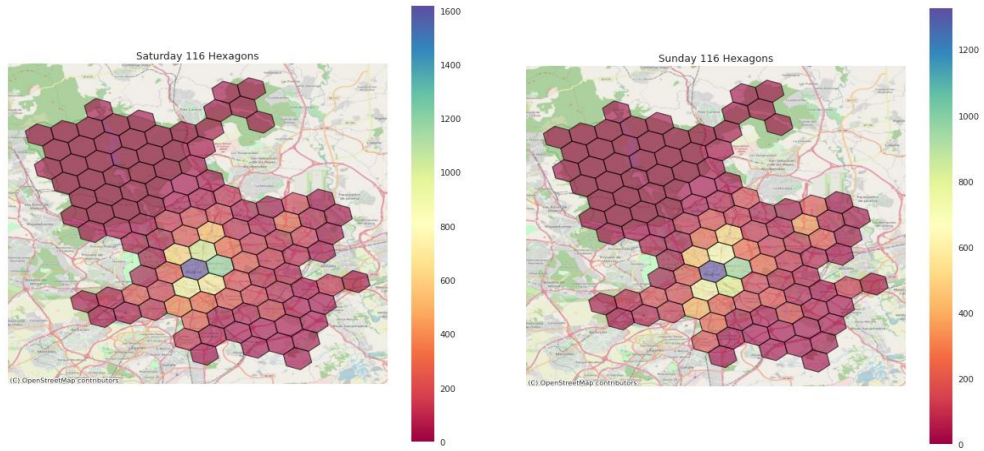
Zonas de bosque o sin acceso por carretera.



4 INTEGRACIÓN Y RESULTADOS

A continuación, se ha repetido este análisis en función del día de la semana.

Día de la semana	Tasa de disponibilidad media
Monday	0,127155
Tuesday	0,131093
Wednesday	0,157231
Thursday	0,128578
Friday	0,138235
Saturday	0,081489
Sunday	0,071197



- Como se puede observar, durante los días laborales la tasa de disponibilidad de taxis es muy similar.
- Durante los fines de semana, esta tasa de disponibilidad cae en picado respecto a los días laborables.
- Adicionalmente, el comportamiento por áreas durante los días laborales es muy similar al descrito en la diapositiva anterior.
- Por otro lado, los fines de semana se observar como la tasa de disponibilidad se contrae también en las áreas interiores y se desplaza la oferta de taxis a unas pocas áreas del centro.

4 INTEGRACIÓN Y RESULTADOS

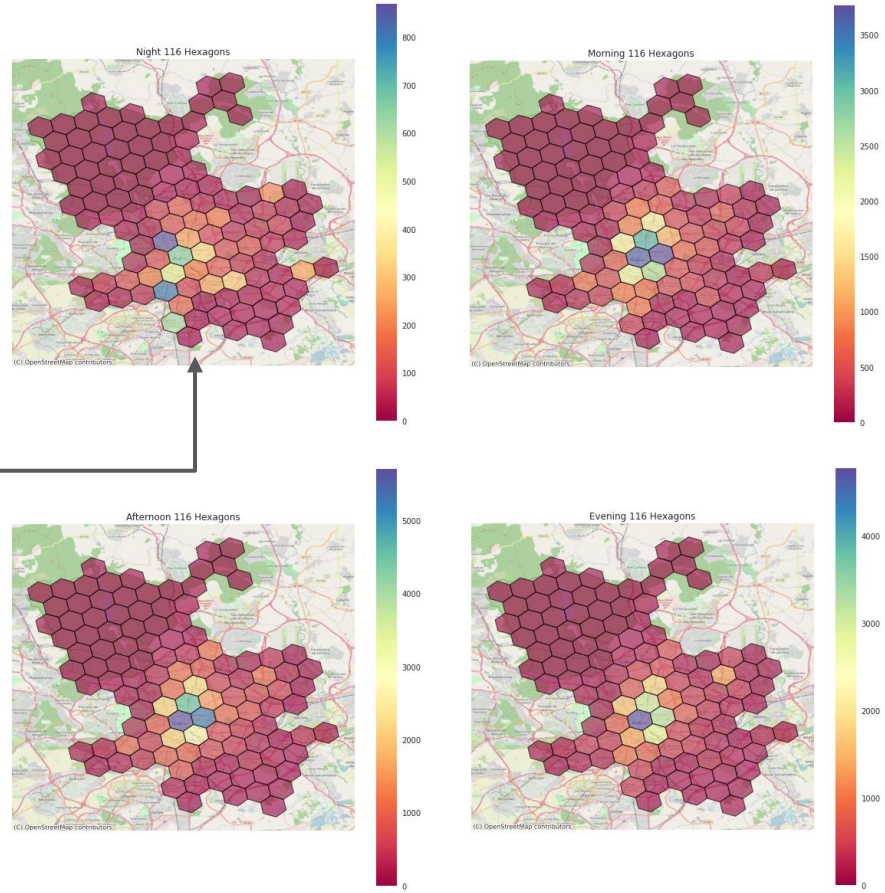
Por último, se ha analizado en función de la hora y el periodo del día.

	Día de la semana	Tasa de disponibilidad
Night	0	0,053287
	1	0,029770
	2	0,023328
	3	0,021102
	4	0,020841
	5	0,033180
Morning	6	0,060392
	7	0,084549
	8	0,121424
	9	0,162775
	10	0,184184
	11	0,197234
Afternoon	12	0,202444
	13	0,200218
	14	0,182858
	15	0,170543
	16	0,165214
	17	0,165238
Evening	18	0,166161
	19	0,158346
	20	0,134805
	21	0,121992
	22	0,107380
	23	0,095514

Durante la noche, desde las 10pm hasta las 7am, la tasa de disponibilidad media cae hasta niveles mínimos.

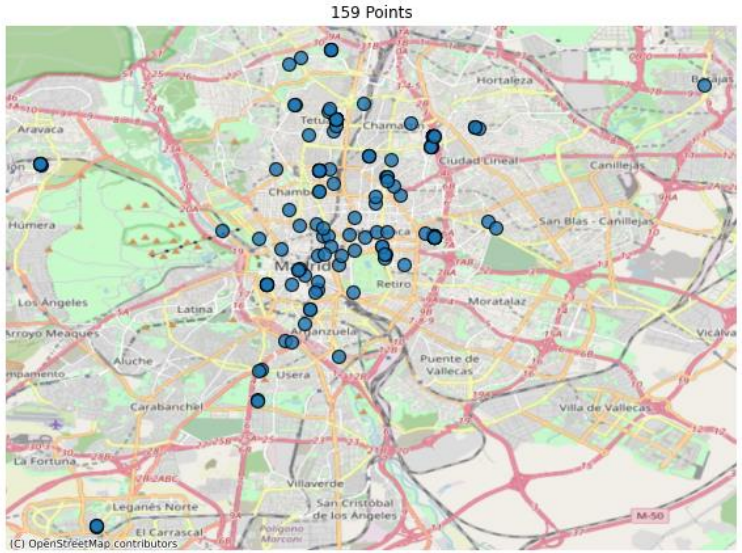
En ese momento (noche), se difuminan las áreas de mayor de disponibilidad y el centro deja de concentrar todos los taxis.

En las horas centrales del día, desde las 9am hasta las 7pm, es cuando la tasa de disponibilidad es mayor



4 INTEGRACIÓN Y RESULTADOS

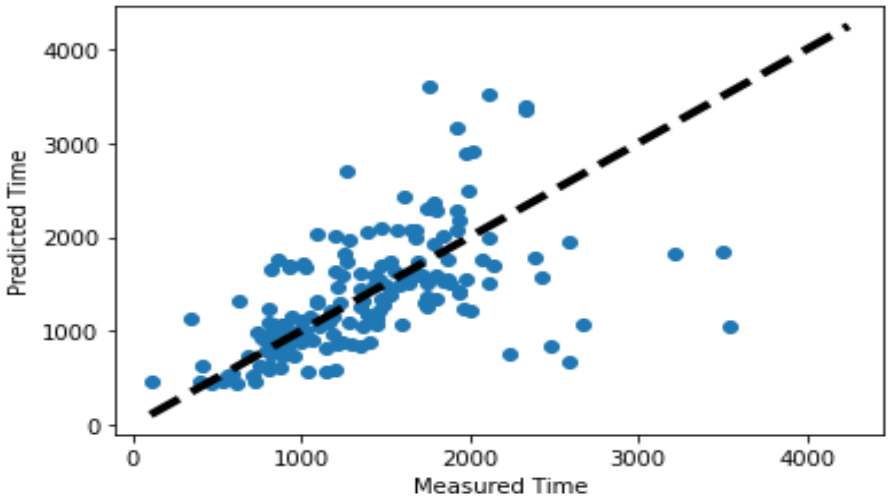
La última parte del proyecto ha consistido en aplicar la matriz, creada a partir de los dos modelos desarrollados, sobre un conjunto de datos reales, el histórico de viajes realizados en taxi.



Como se puede observar en la imagen, las áreas de recogida se distribuyen por diferentes áreas de Madrid.

Esto incluye un total de 159 viajes realizados.

R^2 de 0.063

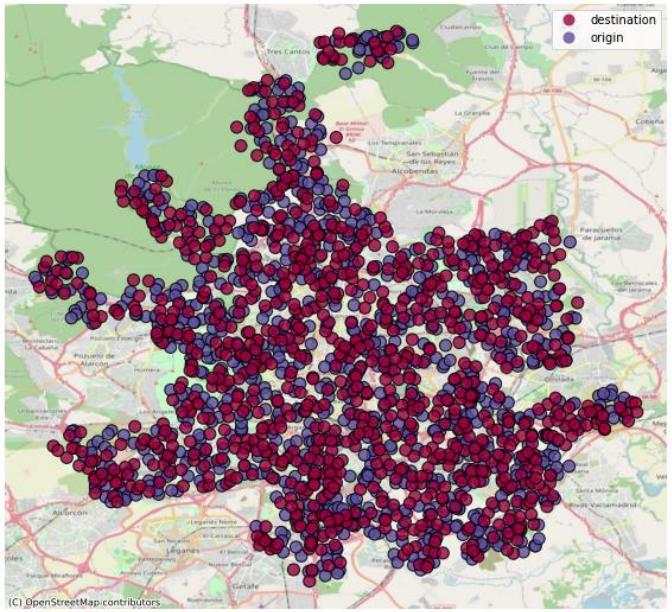


La mayoría de las estimaciones para este conjunto de datos reales se corresponden a la realidad.

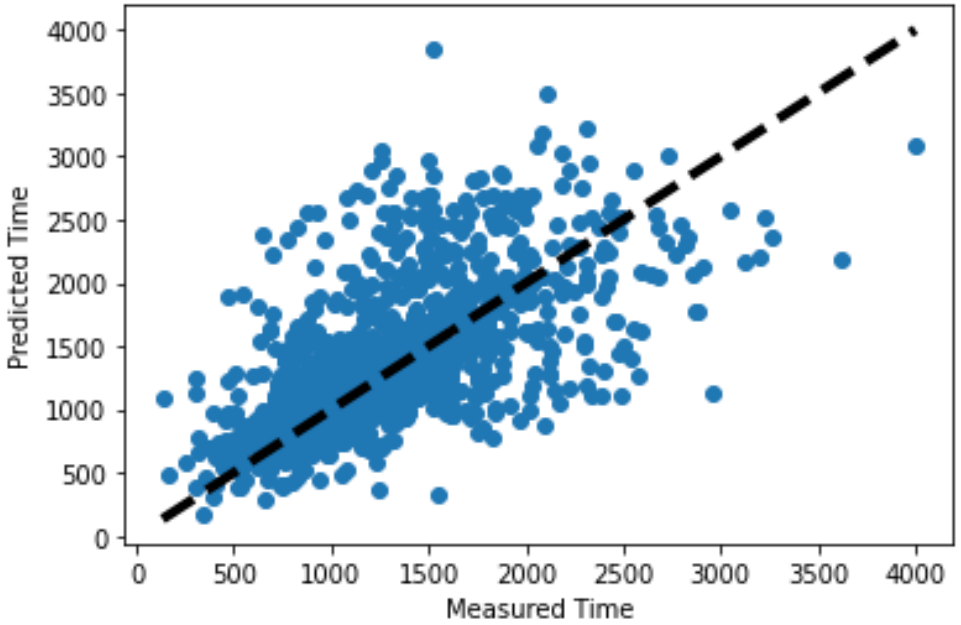
Sin embargo, ha aumentado la tasa de error frente a los resultados obtenidos en la validación individual de los modelos.

4 INTEGRACIÓN Y RESULTADOS

Tras analizar estos resultados, se ha llegado a la conclusión de que el principal problema (y futuro punto de mejora) está en la selección de centroides en los pares origen-destino del histórico de trayectos.



Para validar esta hipótesis, se han seleccionado **1000 puntos aleatorios** en el municipio de Madrid y se ha utilizado la **API de Google para estimar los tiempos** de trayecto.



Finalmente se han comparado estos resultados con los devueltos por el modelo desarrollado para estimar tiempos.
Como se puede comprobar, **la tasa de error a crecido respecto a la validación** del modelo, especialmente en los puntos más alejados del centroide.

5

CONCLUSIONES Y TRABAJO FUTURO

5 CONCLUSIONES Y TRABAJO FUTURO

Tras las realización de este proyecto las principales **conclusiones** obtenidas han sido:

- Las aplicaciones de movilidad cuentan con avanzadas plataformas tecnológicas, sin embargo, no ofrecen al usuario estimaciones del tiempo de recogida antes de que este realice una búsqueda y contacten con los vehículos cercanos.
- Las plataformas *cloud* permiten diseñar un sistema que se adapte a las necesidades y escale los recursos bajo demanda.
- Es posible dividir un territorio en áreas hexagonales para garantizar que todos los vecinos estén a la misma distancia.
- Existen numerosas librerías que simplifican el tratamiento de datos geográficos.
- Los modelos de *ensemble* permiten reducir el error y ofrecen unos buenos resultados en diferentes tipos de problemas.
- Es posible optimizar los modelos aplicando técnicas que iteran sobre las posibles configuraciones hasta obtener la mejor combinación.
- Ha sido posible desarrollar un sistema que devuelva estimaciones aceptables en un tiempo de respuesta mínimo.

5 CONCLUSIONES Y TRABAJO FUTURO

El sistema desarrollado tiene numerosas vías de evolución, es posible explorar diferentes alternativas como:

Aumentar el volumen de datos

Este será un punto crítico para mejorar la precisión de los modelos.

Es necesario aumentar la cantidad de información en ambos *datasets* (disponibilidad y tiempos de trayecto)

Ampliar el histórico disponible

Actualmente se cuenta con un histórico de 3 meses, desde febrero hasta mayo.

Esto será fundamental para capturar ciertos sesgos asociados a la estacionalidad de los datos.

Generar trayectos entre nuevos pares

Actualmente, el dataset de tiempos de trayecto cuenta con información de viajes entre los centroides de las áreas.

Es necesario enriquecer los datos seleccionados puntos aleatorios.

Reducir tamaño de las áreas definidas

Estas abarcan un gran perímetro, reducir su tamaño y ampliar el número de áreas mejorará la precisión de los modelos.

Para ello es necesario disponer de un volumen de datos superior al actual.

Nuevas fuentes externas

Esto incluye información que podría ser muy valiosa para enriquecer los modelos, incluyendo: datos meteorológicos, información del tráfico, calendarios, ubicación de áreas de interés, etc.

Estudiar otro tipo de modelos

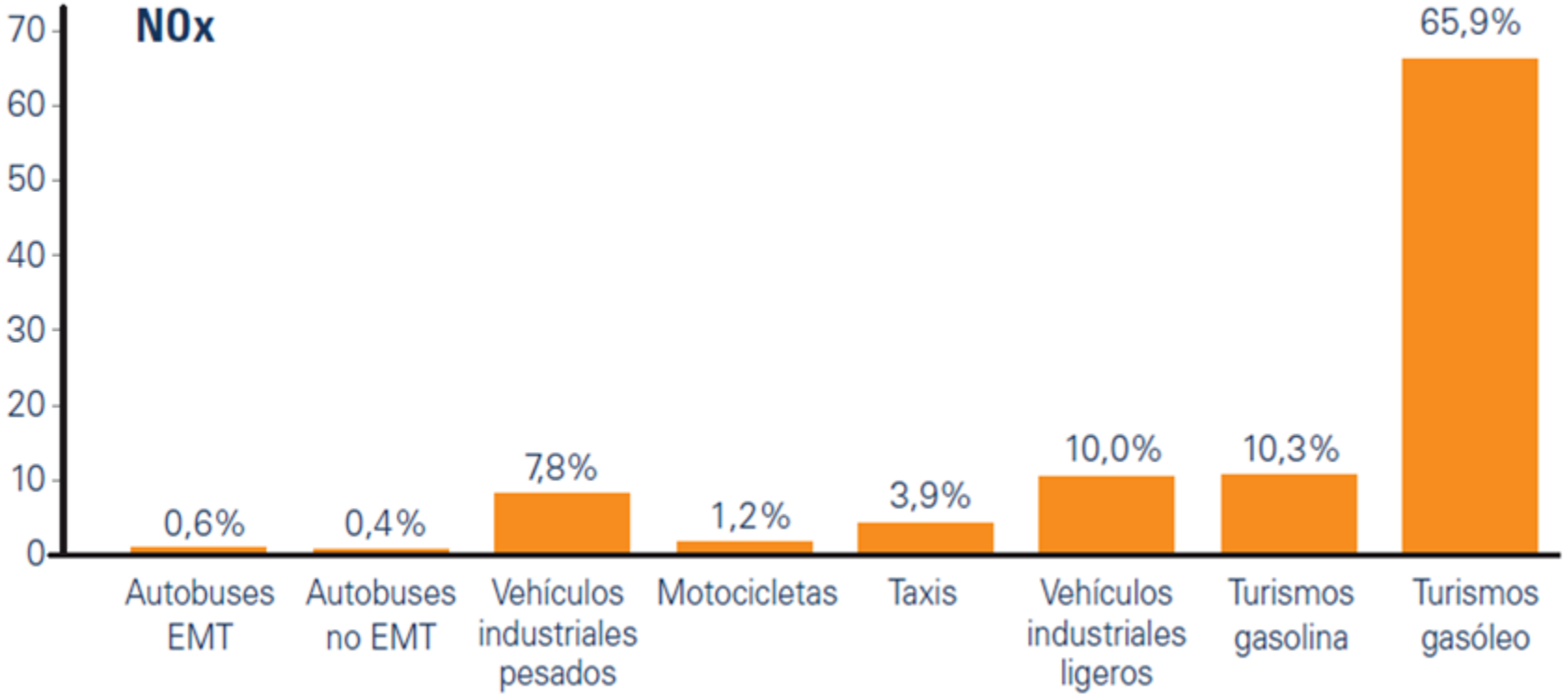
Este proyecto se ha centrado en algoritmos de *ensemble*, concretamente *random forest* y *XGBoost*.

Se podrían estudiar otras alternativas y comparar los resultados con los actuales.

¿PREGUNTAS?

1 INTRODUCCIÓN

"Algunas de las principales preocupaciones de los madrileños, según la Encuesta de Calidad de Vida del Observatorio de la Ciudad, son el tráfico y la contaminación del aire que respiramos"

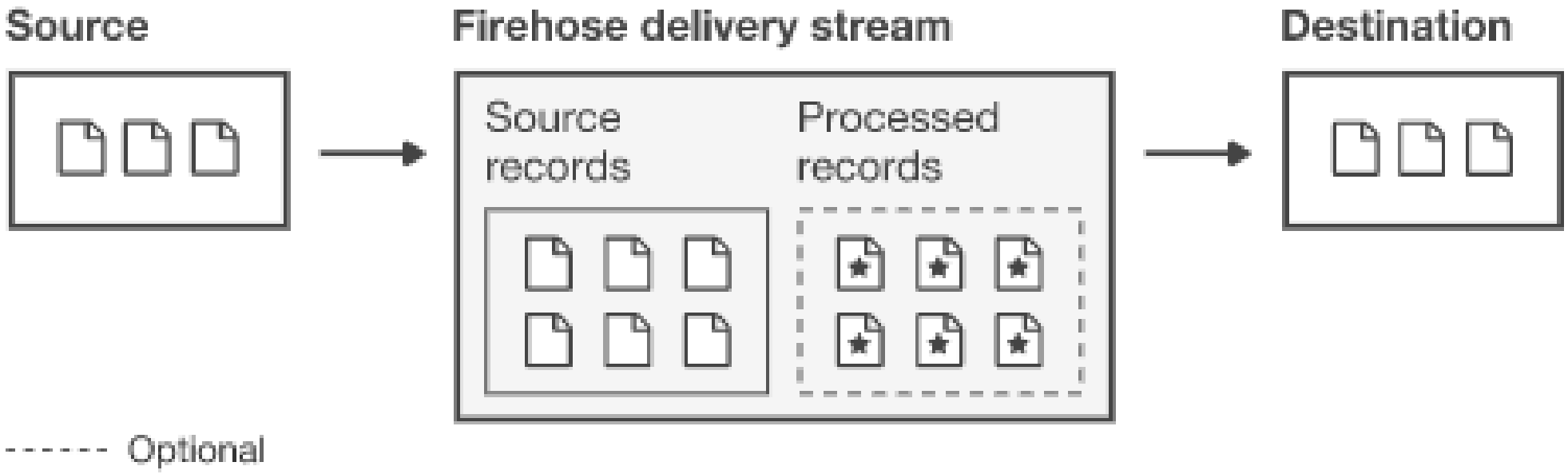


Contaminación de NOx en Madrid producida por el tráfico rodado, teniendo en cuenta los índices de ocupación.

Fuente: Xavier Querol. "La calidad del aire en las ciudades: Un reto mundial"

4 DISEÑO Y DESARROLLO

Recogida y entrega de datos mediante AWS Firehose.

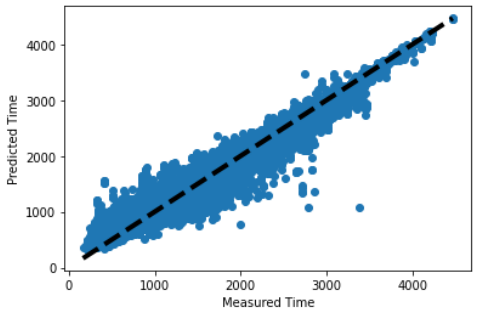


3 DISEÑO Y DESARROLLO

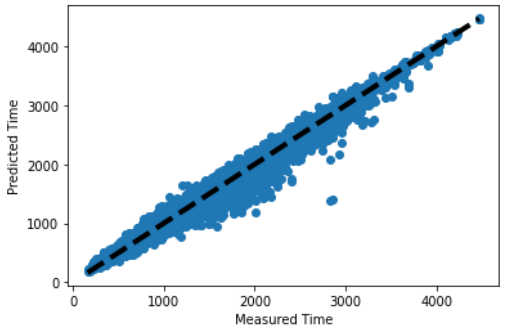
Por último, se han medido los resultados obtenidos por el modelo con el objetivo de validar los resultados del mismo.

En primer lugar, se han comparado los resultados obtenidos con el **modelo optimizado (2)** frente al **modelo inicial (1).**

1



2



Comparando estos resultados se puede validar la mejora en los resultados obtenidos.

Por último se han calculado algunas de las métricas más utilizadas en la evaluación de este tipo de modelos.

```
MSE: 7289.270547101851

RMSE: 85.377224990637

MAE: 57.3726597824806

R2: 0.9875648037079984
```

- **RMSE**: Raíz cuadrada del error cuadrático medio (**MSE**).
- **MAE**: Error absoluto medio o Mean Absolute Error.
- $\boldsymbol{R^2}$: Coeficiente de determinación o coefficient of determination.